



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**DYNAMIC TOWED ARRAY MODELS AND STATE
ESTIMATION FOR UNDERWATER TARGET
TRACKING**

by

Zachariah H. Stiles

September 2013

Thesis Co-Advisors:

Robert G. Hutchins
Xiaoping Yun

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

| | | | | |
|--|---|--|--|--|
| REPORT DOCUMENTATION PAGE | | | <i>Form Approved OMB No. 0704-0188</i> | |
| Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503. | | | | |
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE September 2013 | 3. REPORT TYPE AND DATES COVERED Master's Thesis | |
| 4. TITLE AND SUBTITLE DYNAMIC TOWED ARRAY MODELS AND STATE ESTIMATION FOR UNDERWATER TARGET TRACKING | | | 5. FUNDING NUMBERS N/A | |
| 6. AUTHOR(S) Zachariah H. Stiles | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | |
| 9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A | | | 10. SPONSORING/MONITORING AGENCY REPORT NUMBER | |
| 11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ___N/A___. | | | | |
| 12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (maximum 200 words) <p>The ability of Towed Linear hydrophone Arrays (TLA) to detect submarine-emitted narrow band tonals makes them the submarine tracking sensor of choice. Recent TLA improvements allow surface ships, Unmanned Underwater Vehicles (UUVs), Unmanned Surface Vehicles (USVs), and submarines alike to detect modern submarines by towing arrays. Allowing the full spectrum of Navy assets access into the Anti-submarine Warfare (ASW) arena is vital to countering future submerged threats. The generation of dynamic TLA and state estimation models in Simulink is detailed in this thesis. The dynamic TLA model receives user-specified TLA parameters and performs Dolph-Chebyshev optimization to form a set of beams which are steered for tracking. The TLA parameters can be specified to meet the needs of the towing vehicle, whether it is a submarine, ship, USV or UUV. The state estimation model uses outputs received from a mobile platform towing an array to estimate the target state. The state estimation model uses both bearing-only and Doppler-bearing Extended Kalman Filters to estimate target state. These models provide a basic platform which can be used to enhance ASW capabilities. Specifically, the models can aid in determining optimal future ASW-asset allocation, improving TLA tracking algorithms, and improving information presented to submarine operators.</p> | | | | |
| 14. SUBJECT TERMS Submarine, UUV, USV, ASW, Towed Linear Array, Beamforming, Dolph-Chebyshev Optimization, Bearings-only Tracking, Doppler-bearing Tracking, Extended Kalman Filter, DBT, BOT | | | 15. NUMBER OF PAGES 125 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT Unclassified | 18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified | 19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified | 20. LIMITATION OF ABSTRACT UU | |

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**DYNAMIC TOWED ARRAY MODELS AND STATE ESTIMATION FOR
UNDERWATER TARGET TRACKING**

Zachariah H. Stiles
Lieutenant Commander, United States Navy
B.S., University of Colorado, Colorado Springs, 2003
M.S., Naval Postgraduate School, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
September 2013**

Author: Zachariah H. Stiles

Approved by: Robert G. Hutchins
Thesis Co-Advisor

Xiaoping Yun
Thesis Co-Advisor

R. Clark Robertson
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The ability of Towed Linear hydrophone Arrays (TLA) to detect submarine-emitted narrow band tonals makes them the submarine tracking sensor of choice. Recent TLA improvements allow surface ships, Unmanned Underwater Vehicles (UUVs), Unmanned Surface Vehicles (USVs), and submarines alike to detect modern submarines by towing arrays. Allowing the full spectrum of Navy assets access into the Anti-submarine Warfare (ASW) arena is vital to countering future submerged threats. The generation of dynamic TLA and state estimation models in Simulink is detailed in this thesis. The dynamic TLA model receives user-specified TLA parameters and performs Dolph-Chebyshev optimization to form a set of beams which are steered for tracking. The TLA parameters can be specified to meet the needs of the towing vehicle, whether it is a submarine, ship, USV or UUV. The state estimation model uses outputs received from a mobile platform towing an array to estimate the target state. The state estimation model uses both bearing-only and Doppler-bearing Extended Kalman Filters to estimate target state. These models provide a basic platform which can be used to enhance ASW capabilities. Specifically, the models can aid in determining optimal future ASW-asset allocation, improving TLA tracking algorithms, and improving information presented to submarine operators.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

| | | |
|-------------|--|-----------|
| I. | INTRODUCTION..... | 1 |
| A. | BACKGROUND | 1 |
| B. | OBJECTIVE | 3 |
| C. | APPROACH..... | 4 |
| D. | THESIS ORGANIZATION..... | 4 |
| II. | TOWED HYDROPHONE LINEAR ARRAY ANALYSIS..... | 5 |
| A. | INTRODUCTION..... | 5 |
| B. | TOWED LINEAR HYDROPHONE ARRAY BACKGROUND..... | 5 |
| 1. | Overview | 5 |
| 2. | TLA Outputs | 8 |
| 3. | TLA Limitations..... | 8 |
| 4. | Effects of Altering Physical Parameters of the Array | 9 |
| a. | <i>Hydrophone Spacing.....</i> | <i>10</i> |
| b. | <i>Number of Hydrophones in an Array</i> | <i>10</i> |
| c. | <i>Array Scope</i> | <i>11</i> |
| d. | <i>Hydrophone Element Characteristics</i> | <i>11</i> |
| C. | DYNAMIC TLA MODELING | 11 |
| 1. | TLA Model Setup..... | 11 |
| 2. | Beamforming Scheme for TLA Model..... | 13 |
| 3. | Determining Signal Bearing..... | 16 |
| a. | <i>Beam Steering</i> | <i>17</i> |
| b. | <i>Beam Spacing.....</i> | <i>17</i> |
| c. | <i>Bearing Determination</i> | <i>18</i> |
| D. | TLA MODEL PERFORMANCE..... | 21 |
| E. | CHAPTER SUMMARY..... | 31 |
| III. | TARGET STATE ESTIMATION..... | 33 |
| A. | INTRODUCTION..... | 33 |
| B. | CURRENT TLA TARGET STATE ESTIMATION TECHNIQUES..... | 33 |
| C. | TARGET STATE ESTIMATION VIA EXTENDED KALMAN FILTERS | 34 |
| 1. | EKF Background | 34 |
| 2. | Bearing-Only Tracking Using BO-EKF | 34 |
| a. | <i>Overview</i> | <i>34</i> |
| b. | <i>BO-EKF Model</i> | <i>38</i> |
| c. | <i>Filter Performance.....</i> | <i>39</i> |
| 3. | Doppler-bearing Tracking Using DB-EKF | 47 |
| a. | <i>Overview</i> | <i>47</i> |
| b. | <i>DB-EKF Model</i> | <i>50</i> |
| c. | <i>Filter Performance.....</i> | <i>51</i> |
| 4. | Omission Choice for SNR Information for EKF | 61 |
| D. | CHAPTER SUMMARY..... | 62 |

| | | |
|--------------------|--|------------|
| IV. | MOTION AND ENVIRONMENT MODELING | 63 |
| A. | MODELING TARGET AND RECEIVER MOVEMENT..... | 63 |
| 1. | Platform Motion..... | 63 |
| 2. | Sensor Motion..... | 64 |
| B. | MODELING OCEAN ENVIRONMENT..... | 65 |
| C. | CHAPTER SUMMARY..... | 66 |
| V. | CONCLUSIONS AND FUTURE RESEARCH..... | 67 |
| A. | CONCLUSIONS | 67 |
| B. | FUTURE RESEARCH..... | 67 |
| APPENDIX A. | TLA MODEL INITIALIZATION AND PLOTTING SHELL FILES..... | 69 |
| A. | TLA_SIM.M: TLA SIMULATION SHELL..... | 69 |
| B. | TLA_BEAM_SETUP.M: BEAM SETUP FUNCTION FILE..... | 70 |
| C. | BEAMPATTERN_DC.M: PLOTTING FAR-FIELD BEAM PATTERN USING DOLP-CHEBYSHEV AMPLITUDE AND PHASE WEIGHTING..... | 73 |
| D. | DOLPH_BW.M: CALCULATING BEAMWIDTHS | 74 |
| APPENDIX B. | TLA SIMULINK FILES..... | 77 |
| A. | EMBEDDED MATLAB FUNCTION BLOCK “DYNAMICS” | 84 |
| B. | M_EXPAND.M | 84 |
| C. | BEAM_SUM.M..... | 84 |
| D. | EMBEDDED MATLAB FUNCTION BLOCK “BRNG_TEST” | 84 |
| APPENDIX C. | FILTER_SIM: STATE ESTIMATION INITIALIZATION AND PLOTTING SHELL | 87 |
| APPENDIX D. | TRACKING_SIM.SLX: STATE ESTIMATION SIMULINK FILE .. | 91 |
| A. | EMBEDDED MATLAB FUNCTION BLOCK “EKFKALMANBO” | 95 |
| B. | EMBEDDED MATLAB FUNCTION BLOCK “EXTKALMANV1” (DB-EKF V1)..... | 96 |
| C. | EMBEDDED MATLAB FUNCTION BLOCK “EXTKALMANV2” | 97 |
| | LIST OF REFERENCES | 101 |
| | INITIAL DISTRIBUTION LIST | 103 |

LIST OF FIGURES

| | | |
|------------|--|----|
| Figure 1. | A generic towed linear array layout..... | 5 |
| Figure 2. | Noise source ambiguity present when using a TLA with omnidirectional hydrophones. After [11]..... | 7 |
| Figure 3. | Basic towed array geometrical setup with zero degrees defined along positive X-axis. | 12 |
| Figure 4. | Representation of delay and sum beamforming. After [16]. | 13 |
| Figure 5. | Main lobe steered to 90 degrees formed using Dolph-Chebyshev optimization method. | 16 |
| Figure 6. | Graphical display of beams and angles for a generic seven-element array. | 19 |
| Figure 7. | Plot of beam one to beam two power ratios vs. time used to generate coefficients for endfire bearing determination..... | 20 |
| Figure 8. | A general depiction of how to implement a dynamic TLA model. | 22 |
| Figure 9. | Far-field beam pattern for Case A towed linear array. | 23 |
| Figure 10. | Beam power signals for Case A simulation run..... | 24 |
| Figure 11. | Case A true and measured bearing over time. | 24 |
| Figure 12. | Far-field beam pattern for Case B towed linear array..... | 25 |
| Figure 13. | Beam power signals for Case B simulation run..... | 26 |
| Figure 14. | Case B true and measured bearing over time..... | 26 |
| Figure 15. | Far-field beam pattern for Case C towed linear array..... | 27 |
| Figure 16. | Beam power signals for Case C simulation run..... | 28 |
| Figure 17. | Case C true and measured bearing over time..... | 28 |
| Figure 18. | Beam power signals for Case A1..... | 30 |
| Figure 19. | True and measured bearing over time for Case A1. | 30 |
| Figure 20. | BO-EKF implementation in Simulink. | 38 |
| Figure 21. | Case I BO-EKF simulation trajectories and position estimate. | 41 |
| Figure 22. | Case I BO-EKF simulation results with target range error over time depicted in (A), total range to target over time depicted in (B), and range error as a percentage of total target range depicted in (C). | 42 |
| Figure 23. | BO-EKF simulation Case I bearing error (A) and measured and estimated bearings over time (B). | 43 |
| Figure 24. | Case II BO-EKF receiver and target trajectories with estimated target position..... | 44 |
| Figure 25. | BO-EKF geometry traversing 180 to -180 degree singularity. | 45 |
| Figure 26. | BO-EKF simulation through a singularity point with estimated range error is shown in (A), range to target is shown in (B) and range error as a percentage of total range is shown in (C). | 46 |
| Figure 27. | A depiction of a generic DB-EKF Embedded Matlab Function block. | 50 |
| Figure 28. | Case I DB-EKF and BO-EKF mean estimated target positions with actual target and receiver trajectories. | 52 |
| Figure 29. | Case II simulation for DB-EKF and BO-EKF showing mean range errors (A), range to target platform (B), and range error as a percentage of total range (C). | 54 |

| | | |
|------------|--|----|
| Figure 30. | Case II DB-EKF and BO-EKF bearing error over time (A), and measured and estimated bearings over time (B). | 55 |
| Figure 31. | Case II target and receiver trajectories with mean estimates for target position from DB-EKF and BO-EKF. | 56 |
| Figure 32. | Case IA target and receiver trajectories with mean estimates for target position from DB-EKF and BO-EKF. | 57 |
| Figure 33. | Case IB target and receiver trajectories and mean estimated target positions for DB-EKF and BO-EKF. | 58 |
| Figure 34. | Case IB DB-EKF base frequency error over time. | 59 |
| Figure 35. | True target and receiver trajectories with mean estimated positions using DB-EKF and BO-EKF with biasing. | 60 |
| Figure 36. | True target and receiver trajectories with DB-EKF and BO-EKF mean estimated positions, where biasing is overcome by increasing bearing rate.... | 61 |
| Figure 37. | Target and receiver platform motion modeling. | 63 |
| Figure 38. | TLA.slx Simulink top level view. | 77 |
| Figure 39. | Interior of motion simulation subsystem (TLA/Motion Simulation). | 78 |
| Figure 40. | Interior of target motion subsystem block (TLA/Motion Simulation/Target Motion). | 79 |
| Figure 41. | Interior of array motion subsystem block (TLA/Motion Simulation/Array Motion). | 79 |
| Figure 42. | Interior of velocity in LOS subsystem block (TLA/Velocity in LOS). | 80 |
| Figure 43. | Interior of the element position simulation subsystem block (TLA/Element Position Simulation). | 81 |
| Figure 44. | Interior of spherically attenuated signal received at elements subsystem block (TLA/SASRaES). | 82 |
| Figure 45. | Interior of beam selector subsystem (TLA/Beam Selector). | 83 |
| Figure 46. | Top level view of Tracking_Sim.slx. | 91 |
| Figure 47. | Interior of perfect outs subsystem block (Tracking_Sim/Perfect Outs). Note: Target and Array Motion Subsystems have the same interior contents as the Target Motion and Array Motion Subsystems in the TLA.slx model. | 92 |
| Figure 48. | Interior of measured outs subsystem block (Tracking_Sim/Measured Outs). | 93 |
| Figure 49. | Interior of filter subsystem block (Tracking_Sim/Filter). | 94 |

LIST OF TABLES

| | | |
|----------|---|----|
| Table 1. | Parameters used for TLA test cases. | 22 |
| Table 2. | Maneuvering receiver simulation Case A1 parameter setup. | 29 |
| Table 3. | Simulation parameters used for Case I and Case II simulation runs. | 40 |
| Table 4. | Simulation parameters used for Case I BO-EKF simulation run. | 40 |
| Table 5. | Case I simulation parameters for DB-EKF and BO-EKF comparison. | 53 |
| Table 6. | Case I trajectory description for DB-EKF and BO-EKF comparison. | 53 |

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---------|--|
| ASW | Anti-submarine Warfare |
| BO-EKF | Bearing-only Extended Kalman Filter |
| CONOPS | Concept of Operations |
| DB-EKF | Doppler-bearing Extended Kalman Filter |
| EKF | Extended Kalman Filter |
| KF | Kalman Filter |
| SNR | Signal to Noise Ratio |
| SSP | Sound Speed Profile |
| SURTASS | Surveillance Towed Array Sonar System |
| TLA | Towed Linear (Hydrophone) Array |
| USV | Unmanned Surface Vehicle |
| UUV | Unmanned Underwater Vehicle |

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The United States Navy currently enjoys maritime superiority in the undersea domain [1]. The Navy promulgates Anti-submarine Warfare goals that they deem necessary to meet to maintain this advantage [2]. One of the reasons that the Navy enjoys this superiority is that it is able to use Towed Linear hydrophone Arrays (TLAs) to track submerged threats. Towed arrays allow the towing platform to detect quiet submarines by their emitted narrow band tonals, which can travel great distances underwater. Unlike radar, passive TLAs do not enjoy constant improvement from commercial use, therefore, most improvements in TLAs are driven by defense applications. When comparing passive TLA technology to a similar technology like radar, the lack of commercial applications and difficult ocean environment have led to a slower rate of improvement. Because TLAs are suited for detecting submerged threat contacts, improving the arrays themselves, the methods used process their outputs, and the TLA tracking tactics may be of large importance to the Navy moving forward. Specifically, the following steps should be considered:

- Improving the state estimation algorithms used to process TLA outputs.
- Improving the utility of the information displayed to operators using TLAs to track threat contacts by:
 - Enabling operators to determine the uncertainty of the best estimate for threat contact at all times.
 - Providing operators with dynamic decision aids to recommend most mathematically sound maneuvers to best safely track the threat contact.
- Expanding Anti-submarine Warfare capabilities of non-submarine assets in the U.S. Navy portfolio. The expanded capabilities should specifically address Surface Combatants, Unmanned Underwater Vehicles and Unmanned Surface Vehicles usage of TLAs.

The steps above represent long-term TLA-specific improvements that can be made to improve U.S. ASW capabilities. The overall objective of this thesis is to lay a portion of the framework necessary to accomplish these TLA-specific goals. The overall objective of the thesis was accomplished by:

- Developing a dynamic TLA model used to simulate TLAs with varying sizes and parameters.
- Outlining a method to take physical array parameters for varying arrays and determine an algorithm used to process the incoming array signals into useful outputs.
- Generating a model that passes generic TLA outputs for a specified tracking situation, with added noise, used to test state estimation filters.
- Determining the feasibility of bearing-only Extended Kalman Filters and Doppler-bearing Extended Kalman Filters used for state estimation in passive TLA tracking scenarios.

For the TLA model, the user specifies TLA parameters, and then Dolph-Chebyshev Optimization is used to determine amplitude weights used to form the far-field beam pattern desired for a single beam steered perpendicular to the array. Next, the beamwidth of the single beam is calculated, and additional beams are steered to either side of the beam at the 3 dB down points. Additional beams are spread until the beams are spread from 0 degrees to 180 degrees. The composite far-field beam pattern of all beams to be processed for the array are then presented to the user to verify that desired behavior is met without the presence of grating lobes. The model then simulates a TLA in a tracking scenario assuming generated amplitude and phase weighting arrays and given specified parameters for the motion of platforms, target emitted noise, and ocean environment.

In order to determine if the TLA model worked well, three arrays were simulated, covering a large range of array sizes. A small seven element array, a medium 21 element array and a large 100 element array were simulated in a scenario where the target contact swept from the forward beam to aft beam over the course of an hour. In order to determine if the array worked well during maneuvers, the seven-element array was subjected to a simulation where the towing platform maneuvered during target tracking.

For the state estimation model, noisy output from a TLA is modeled and passed to various filtering blocks used to estimate target state. Two specific filters were analyzed, the bearing-only and Doppler-bearing Extended Kalman Filters. Various scenarios were simulated to determine if the filters were acceptable for target tracking in the undersea environment using passive TLA outputs. Minimal towing platform maneuvering

scenarios were simulated along with scenarios where the towing platform out-maneuvers the target platform. The scenarios were simulated with the true target state used for the initial state estimation as well as erroneous initial state estimates. The overall range in the scenario was also varied to determine if any biasing conditions existed. With these simulations, the good and bad aspects of the bearing-only and Doppler-bearing Extended Kalman filters were identified.

This research demonstrated that the dynamic TLA model is useful when determining what array parameters to use with a specific towing platform. The run time of the model was a limiting factor that precluded its direct connection with the state estimation model. It is recommended that once a satisfactory array configuration is identified that the TLA model be made specifically for that configuration so that the simulation run time can be reduced. Non-dynamic towed array models could be paired with the state estimation model, and control loops based upon estimated target state could be derived. The state estimation model proved that the bearing-only Extended Kalman Filter had too many problems to be used for underwater target tracking. The Doppler-bearing Extended Kalman Filter showed more promise, but some adjustments to the filter, as modeled in this thesis, are necessary for good functionality in undersea target tracking applications.

Vast quantities of work are still required to meet the Navy's future goals for Anti-submarine Warfare. The initial framework to improve the TLA-specific aspects of Anti-submarine Warfare has been developed; the main objective of the thesis was met.

LIST OF REFERENCES

- [1] M. J. Conner. Sustaining undersea dominance. *Proceedings Magazine*, vol. 139/6/1,324. [Online]. Available: <http://www.usni.org/magazines/proceedings/2013-06/sustaining-undersea-dominance>, June 2013
- [2] Task Force ASW. *Anti-submarine warfare: Concepts of operations for the 21st century*. [Online]. Available: <http://www.navy.mil/navydata/policy/asw/asw-conops.pdf>, [Accessed Aug. 2, 2013].

ACKNOWLEDGMENTS

I would like to thank my amazing wife, Shelley, and my wonderful children, Adam and Caroline, for their support and patience while I was working on my thesis and absorbed in my studies. Without your support, love, and care I could have not accomplished all that I have. My days of not being able to play because I have to “do homework” are finally coming to an end. I love each of you with all my heart.

I would like to thank Dr. Xiaoping Yun and Dr. Robert G. Hutchins for their amazing tutelage both inside and outside of the classroom. I have learned a great deal, extending beyond what this thesis addresses. They were patient and continually available to direct or redirect my research. Most importantly, they both consistently reminded me of the ultimate goal, which was to learn something.

I also received guidance, assistance and insight from Dr. Lawrence Ziomek, Dr. Timothy Chung, and Dr. James Calusdian. They helped point me in the right direction many times and for that I am very grateful.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The United States Navy is currently enjoying maritime superiority in the undersea domain [1]. Maintaining this crucial advantage is a key tenant of United States' Anti-submarine Warfare (ASW) concept of operations (CONOPS). The ASW CONOPS provide two key operational level goals [2]:

Hold Enemy Forces at Risk: We will deny enemy submarines an offensive capability by maintaining the ability to destroy them, if and when required, at a time and place of our choosing.

Secure Friendly Maneuver Area: We will drive away or destroy enemy submarines, thereby protecting maritime operating areas. We will protect US and coalition naval combatants, support ships, and merchant shipping from undersea attack within and enroute to vital operating areas.

In order to hold enemy forces at risk we must be able to track them and have assets in the general vicinity in order to destroy them, if deemed necessary. Historically, the best weapon for submarine tracking and prosecution is a friendly submarine. The demand signal for submarine mission tasking by Combatant Commanders has been increasing dramatically over time as the unique capabilities of the submarine platform have been realized [1]. At the same time, the number of submarines in the fleet is decreasing [3]. Recent proliferation of enemy submarine technologies and a general increase in the number of submarines in the inventories of the various countries of the world point to a problem that cannot be ignored. Submarine tracking and prosecution tactics must be practiced and improved over time or the advantage in this arena will surely disappear. Concurrently, it is necessary to enable other Navy combatants access to fight and defend in the ASW arena to take some of the burden off of the submarine force so that it can continue to provide unique mission capabilities to Combatant Commanders.

The undersea environment provides a large advantage to platforms that can use it. Historically we note that old, noisy and unsophisticated submarines can cause problems for surface combatants and go undetected entirely from air assets. This lesson has been re-learned recently by the intentional surfacing of previously undetected submarines close

to an aircraft carrier in the middle of a carrier group [4] and the sinking of a South Korean corvette by an undetected North Korean submarine, while the U.S. and South Korea were actively engaged in ASW exercises in the area [5].

The first step in holding enemy submarines at risk or securing a friendly maneuver area is detecting and then tracking the enemy submarine. This is best done by exploiting narrow band acoustic signals put into the water by submarines. Newer submarines are able to effectively limit the amount of broad band noise that they emit, therefore, broadband noise detection and tracking is less useful than narrow band noise detection and tracking. Narrow band acoustic tonal tracking is best accomplished using hydrophone arrays that are towed behind a submarine, surface ship, Unmanned Surface Vehicle (USV), or Unmanned Underwater Vehicle (UUV). Fixed hydrophone arrays which are placed at the bottom of the ocean are also very effective at submarine tracking but are expensive, hard to maintain and are easily foiled. Linear towed hydrophone arrays (TLA) are typically used to detect and track submarines and can generally be used actively or passively. Active TLA usage is typically limited to Surveillance Towed Array Sonar System (SURTASS) ships. Passive TLA usage is usually employed on submarines and surface combatants [6].

Passively tracking with TLAs in the complex ocean environment is a difficult task that is still being refined today. Unlike radar tracking, which has many commercial and non-military applications, narrow band undersea tracking algorithms and tactics have evolved more slowly over time. This has resulted in algorithms that are less efficient than they could be, while at the same time providing submarine operators with overly complex and confusing outputs that make interpretation difficult [7].

In order to improve submarine-based ASW capabilities and meet the future goals of the ASW CONOPS it would be beneficial to:

- Improve narrow band tracking algorithms.
- Improve usefulness of information displayed to submarine operators.
- Provide submarine operators with dynamic decision making aids to include:

- Dynamically determining best tracking position based upon own ship noise, target state uncertainty, expected target noise emission profile, and current environmental conditions.
- Dynamically outputting optimal control solution to safely proceed to desired best tracking position taking into account target state uncertainty, loss of tracking ability after a maneuver, desired maximum range rate, and minimum range for safety of ship and counter-detection.
- Geographically displaying an indication of the uncertainty in the target state estimate.
- Enable trial maneuvers to be entered and warn operators when an unsafe trial maneuver is input, taking into account minimum possible range, worst target maneuver occurring when own ship array becomes unstable and target increasing speed to maximum non-cavitation speed.

It is necessary to improve ship, air, USV, and UUV ASW capabilities to effectively neutralize the advantage provided by the undersea environment to submerged threats. Therefore, it would be beneficial to accomplish the following steps to aid in meeting the future goals of the ASW CONOPS:

- Deploy USV or UUV assets that trail TLAs and have broadcast capabilities. Use these assets to create submarine detection fences around harbors, high value assets like carrier groups, or enemy transit zones.
- Develop control algorithms that allow USVs and UUVs to track threat submarines upon detection and report findings to centralized ASW asset.
- Develop deployment tactics and determine logistics support necessary to deploy unmanned ASW assets in large areas.
- Determine how to convey submerged target information from unmanned ASW assets to surface and air components of the Navy. This would enable them to engage or avoid the target platform.

Improving the ASW capabilities of both submarine and non-submarine assets will enable the U.S. Navy to meet the future goals of the ASW CONOPS.

B. OBJECTIVE

The long-term goal is to enable the improvement of U.S. ASW capabilities for both submarine and non-submarine assets so that the U.S. Navy can meet its stated future ASW CONOPS goals. In order to accomplish this goal, the objective of this thesis is to

provide an easily accessible and configurable test platform, in the form of a Simulink Model, where detection and tracking algorithms can be compared and contrasted, ASW best practices can be determined, and CONOPS for unmanned assets utilized in the ASW realm can be developed.

C. APPROACH

The generation of a configurable TLA model is first analyzed with a detailed look at how to process good outputs from the array. Next, using the outputs that can be generated from a TLA, two state estimation filters were identified and evaluated for their use in submarine tracking. The two state estimation filters evaluated were the Bearing-only Extended Kalman Filter (BO-EKF) and the Doppler-bearing Extended Kalman Filter (DB-EKF). Target and receiver platform motion generation and control are then detailed. Next, the specifics as to how the ocean environment is modeled are examined.

D. THESIS ORGANIZATION

The generation of the configurable TLA model and how to process output signals from the various hydrophone outputs is covered in Chapter II. Towed linear array parameter selection, the effects of the various parameters that can be configured, and how to ensure that no grating lobes will be present are also covered. How to generate a target state estimation given the inputs from the TLA is detailed in Chapter III. The BO-EKF and DB-EKF are both modeled and simulated to show their effectiveness at submarine tracking. Target and receiver motion generation and control, as well as how the transmitted target noise is affected by the ocean environment, is explained in Chapter IV. Conclusions and recommendations for future work are contained in Chapter V. The Matlab code and Simulink models used to simulate the TLA and filters are included in Appendices A, B, C and D.

II. TOWED HYDROPHONE LINEAR ARRAY ANALYSIS

A. INTRODUCTION

A first step toward improving ASW capabilities is to develop a working configurable TLA model to be used as a test platform. The current uses and limitations of TLAs are detailed in this chapter. How to make good parameter selections for a desired TLA is also examined. The manipulation of received target acoustic noise into the bearing to the emitting target contact is detailed.

B. TOWED LINEAR HYDROPHONE ARRAY BACKGROUND

1. Overview

TLAs have existed in various forms since 1917 [6]. The earliest TLAs were used to detect submarines, incoming torpedo threats, and to provide visibility to the aft blind spot for submarines, commonly called baffles. A generic figure depicting what a TLA setup looks like is shown in Figure 1.

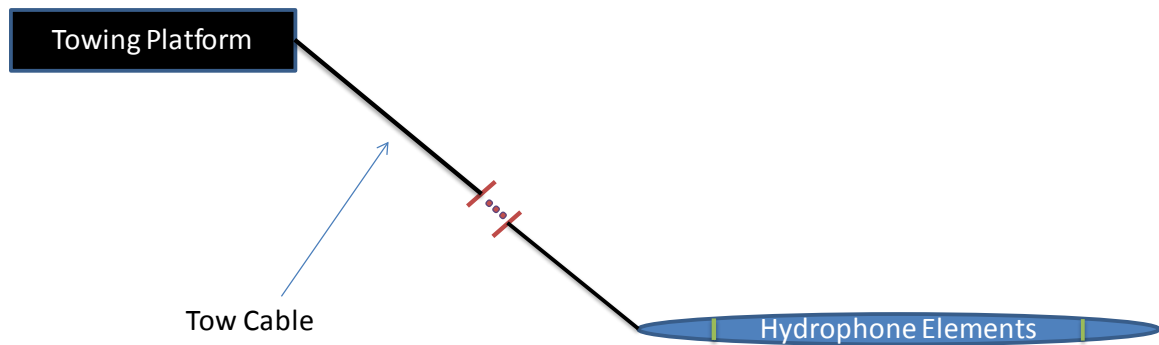


Figure 1. A generic towed linear array layout.

The array tow cable allows the separation of the hydrophone elements from the towing platform, which is generally producing noise. The hydrophone placement, away from the towing platform, enables lower levels of acoustic energy to be detected. When attempting to detect, classify, and track submarines, any method that allows one to

discern smaller incoming acoustic signals from noise is very important as submarines are in general very quiet and hard to detect. By placing the hydrophones in a long line, it allows one to detect low frequency signals which can travel very long distances in water. Typically, any submerged vehicle emits broad-band noise and narrow band acoustic tonals. Lower frequency acoustic tonals can travel great distances in water and can be picked up and used to track a contact of interest.

Tracking submarines is usually something that is done in a passive manner. Tracking passively allows the tracking platform to avoid counter-detection. Active tracking methods, such as the use of an active sonar transmitter, will generally alert the submerged contact at larger ranges than necessary for the tracking platform to gain useful data. This could lead to the tracking platform either being prosecuted or avoided. In addition to counter-detection risks, one can generally glean more useful information from a platform if it is unaware that it is being tracked.

The commercial industry, mostly consisting of the oil industry, uses active towed linear arrays. There have been many advances with these arrays, but in general, they are not useful outside of a mission set like the one performed by SURTASS [6]. Limited commercial applications for passive TLAs mean that information on processing incoming acoustic signals and tracking algorithms is scarce compared to a similar sensor with commercial uses like radar. While radar tracking has developed rapidly, with new techniques being explored on an almost continual basis, towed array tracking underwater has progressed more slowly.

Lack of commercial applications is not the only hindrance to processing techniques and tracking algorithms. The ocean environment presents unique challenges that are either not found in the above water realm or not found to the same extent [8].

The physical setup of the linear hydrophone array also adds some complication. Radar and sonar from spherical sonar domes can maintain visibility on incoming signals when the platform carrying them maneuvers. This is not typically the case for TLAs. In order to maintain track through a turn, a TLA requires information on the geometrical shape of the array. There are new arrays coming out that possess some capability in this

respect, but the TLAs employed by navies in general do not have the ability to track through turns [9]. There is typically little or no information on the state of the array aside from the scope length, tension, and the various outputs: target bearings, received target frequencies, and signal-to-noise ratio (SNR) of received target signals. The lack of state information on the array (i.e., depth, relative positions of elements, cant angle) itself, means that it is not possible to track targets when the tracking ship is turning or anytime that the array becomes unstable (i.e., non-linear). This usually means that data that is passed to a target state estimation algorithm is manually started and stopped, depending upon the perceived state of the array. The lack of data, which can be significantly long in duration and occurs during and after the receiving platform maneuvers, makes target state estimation a difficult task.

The hydrophone elements themselves are typically omnidirectional in nature for most TLAs in use; although, there are newer arrays that use directional hydrophone elements [9]. Omnidirectional element usage results in a TLA that cannot discern an exact bearing to a target in three-dimensional space. Instead, an ambiguous conical angle containing the target contact can be determined. An example, from [10], of what conical angle ambiguity looks like is shown in Figure 2. Maneuvers, or additional information from another source, are necessary to remove directional ambiguity.

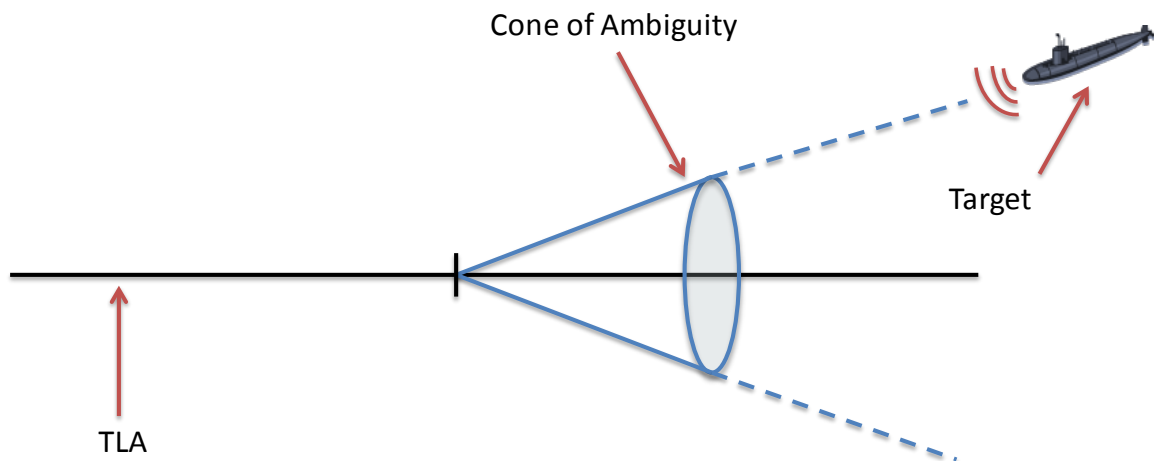


Figure 2. Noise source ambiguity present when using a TLA with omnidirectional hydrophones. After [11].

Using TLAs, one can determine much about the target platform. What can be discerned and how to go about it are discussed in the following sections.

2. TLA Outputs

Towed Linear Arrays using omnidirectional hydrophone elements make it possible to determine the conical angle to the noise source, the narrow band frequency of the received signal, and the SNR of the incoming signal. A knowledgeable sonar operator can sometimes determine signal arrival path, speed of the contact, classification of the contact, and a basic range to the contact if they are in receipt of multiple copies of the same signal from various paths. In general, if a sonar operator could determine all of this information easily each time a contact is gained, it would not be necessary to continue to study this problem. Since submerged contacts do not readily give away all the information necessary at once, in order to determine all the desired information about the contact, it is necessary to study this problem. It is more realistic to expect the receiving platform to be able to classify a contact in a very broad sense, like how many screws and blades the contact has. This information limits what kind of contact category the target platform fits into. For contact speed the same is true, a range of likely speeds can be guessed, but an exact speed of a contact must usually be determined. The range to the contact is generally the hardest piece of the puzzle to track down, as will be seen in Chapter III.

To summarize, the outputs from a TLA that are of interest to generate and then pass to state estimation algorithms are ambiguous conical angle, received frequency, and SNR. Other useful information is at times available from TLAs, but to use them in state estimation algorithms, at this point, results in significant added complications with little or no gain.

3. TLA Limitations

When planning to use TLAs for target tracking, one must take into account some notable limitations. The bearing output from TLAs is ambiguous, as previously stated. Multiple legs of information are necessary to determine range to the target, unless multiple paths of the same signal are received simultaneously. Even if multiple instances

of a signal are received from different paths, it usually takes an operator to verify that the received signals, occurring on separate bearings, are indeed from the same source. Bearings are derived from incoming acoustic sources by assuming that the linear array is in a linear geometry. If the tracking or receiving platform maneuvers, the basis for determining the bearing to the target platform will not be met. Not matching expected array geometry results in output data from the array being erroneous. This means that maneuvering with an array lacking state information on array elements requires a scheme to suppress the data that is processed while the array is unstable.

Historically, passive towed arrays were not able to achieve close to the same level of listening ability as a sensor located on a submarine. Recent progress has enabled surface ship towed arrays to achieve comparable listening ability to modern submarines, when towing less than 13 knots [9]. This means that surface ships, USVs and UUVs [12] could tow arrays and gain enough information to be very useful in the ASW arena.

When deploying a towed array, towing vehicles have to be mindful not to exceed the maximum tension that the array is rated for and to operate at speeds which do not limit its usefulness due to flow noise. The towing vehicle also needs to ensure that it does not drag the towed array on the ocean bottom, which limits very long arrays usage in shallow water areas.

For the purposes of this thesis, the only limitations that are of real concern are TLA inability to track through turns, little or no target range data provided by the TLA, and bearing ambiguity associated with omnidirectional hydrophone elements.

4. Effects of Altering Physical Parameters of the Array

Surface ships, USVs, UUVs, and submarines will, more than likely, use different arrays that are suited for their operating areas and towing conditions. The geometric and physical characteristics of a towed array drive how to best process the incoming acoustic signals into useful outputs. The various physical parameters of the TLA and what the effect is if it is changed are discussed in this section.

a. Hydrophone Spacing

The distance between hydrophone elements d is typically given in units of the wavelength λ of the signal being processed. It is important to note that the wavelength is dependent on the speed of sound for the conducting medium. In the case of sea water, the speed of sound is not a constant but varies with conditions. Therefore, when determining what hydrophone element spacing to use, a limiting speed of sound case should be chosen. The model, developed later in the thesis, uses a constant speed of sound and only one incoming frequency emitted from the target contact.

The closer the elements are together the higher the frequency the array can process without the presence of grating lobes. Grating lobes are side lobes with a magnitude that approaches the magnitude of the main lobe, and they occur when $d \geq \lambda_{\min}/2$ [13]. The presence of grating lobes results in erroneous estimation of incoming acoustic signal direction if not accounted for. Signals with frequencies higher than what can be processed without producing grating lobes are ignored in this study. The hydrophone spacing also affects the directivity of the array via the array factor [14]. The larger the element spacing, the better the directionality of the array will be up to the point where grating lobes occur.

Spacing each hydrophone element at equal distances allows further simplifications if complex weights also satisfy conjugate symmetry [13]. For the purposes of this thesis, equal spacing is used and conjugate symmetry is maintained for weighting factors.

b. Number of Hydrophones in an Array

The number of hydrophones in an array determines how long the array can be given a specific desired spacing. The length of the active portion of the array determines how low the low frequency range of the array is. This means, given specific array spacing, that the number of elements determines the frequency range that the array can process. The number of elements in the array also directly affects the gain that can be achieved by the array [15].

c. Array Scope

The array scope defines how far behind the towing platform the array is located. As the scope gets longer, the array is separated further back from the platform. The further back the array is from the towing platform, the less platform noise that the array is subjected to. As the array scope increases, so does the difficulty in keeping the array in the desired depth stratum and off the ocean bottom. Large array scopes can also cause high tensions on tow cables and induce vibrations that can increase the self-noise of the array.

Conical angles reported from a TLA are with respect to the center of the array's active region. Therefore, if the array is towed at a distance that is not negligible when compared to the distance to a contact that is being tracked, it must be taken into account or the state estimate will be biased.

d. Hydrophone Element Characteristics

Hydrophones can be characterized by the type of directionality they have. The most common type of hydrophone used in TLAs is omnidirectional. Newer arrays utilize directional hydrophone elements [9], which is nice because they can eliminate directional ambiguity problems. Directional hydrophones are not currently used as frequently as omnidirectional hydrophones. Therefore, TLAs used in this thesis are assumed to be omnidirectional.

C. DYNAMIC TLA MODELING

1. TLA Model Setup

To model a dynamic TLA, it is first necessary to have the user specify the physical parameters of the desired array. For the TLA model being presented, the user must specify the distance between elements, in units of base frequency wavelength, the number of hydrophone elements, base frequency of interest, and desired side lobe reduction level. The simulation step size, length, and starting positions and velocities must also be specified. With this information, the model assumes fixed hydrophone element spacing and generates the appropriate Dolph-Chebyshev amplitude weighting

necessary to form a main lobe off the array's broadside. The model then determines a set of phase shifts used to best space the main lobes in bearing for incoming signal direction determination. In order to ensure that the prepared amplitude and phase weighting scheme will work for direction determination later in the model, the setup portion of the model outputs a plot of the far-field beam pattern to the user. With the far-field beam pattern plot, the user can screen for grating lobes or unexpected array behavior. The setup script passes the amplitude and phase weighting data to the Simulink model where it is used to process the incoming acoustic signals on the various beams. The length of the array, which is derived from spacing and number of hydrophone elements, is used to model the physical locations of each hydrophone element in time as described in Chapter IV. The array is assumed to lie along the X-axis, with the beam of the array occurring at 90° and 270° . The angle that a main lobe is steered from the positive X-axis is Ψ .

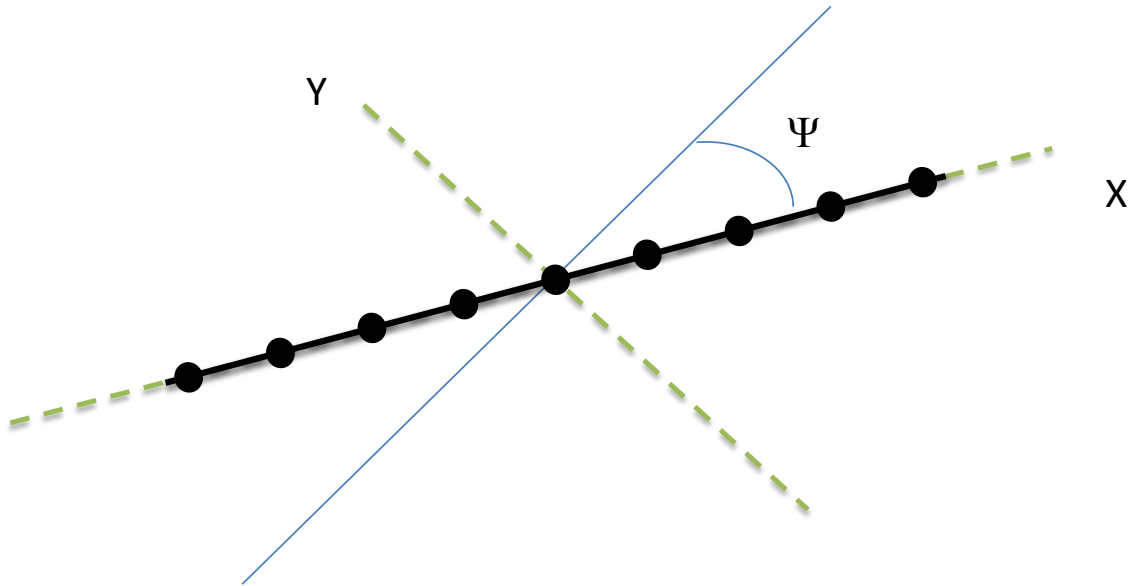


Figure 3. Basic towed array geometrical setup with zero degrees defined along positive X-axis.

The TLA setup script file is contained in the Appendix A. The specifics of how the beams are formed are discussed next.

2. Beamforming Scheme for TLA Model

Beamforming is used to take the inputs from the hydrophone elements and detect an acoustic signal that can be processed for bearing and frequency. Amplitude weights are used to make a prominent main lobe that can be steered. Steering of the array can occur via frequency shifting of the incoming signal or by time delaying the incoming signal; each method ends up accomplishing the same thing [13], [16].

An incoming acoustic signal is received at multiple hydrophones on the array. The hydrophone elements are separated in space, and the signal is received by each hydrophone at separate times. The signals from the elements are delayed and weighted in a manner to stack the signal as if it were coming from a specified angle the beam is steered to. Multiple main lobes or beams of the array are processed at the same time with the same amplitude weights and differing phase weights. Varying the phase weights allows one to effectively steer the individual beams to different bearing so all the beams are processed. The one with the signal arriving from the true source angle results in significantly more power than the other beams, which are not steered toward the incoming signal. A pictorial representation of one main lobe's delay and sum beamforming is shown in Figure 4. With this method, multiple hydrophones enable one to detect what cannot be detected by one hydrophone alone. This method also allows one to effectively determine the direction of the incoming acoustic signal.

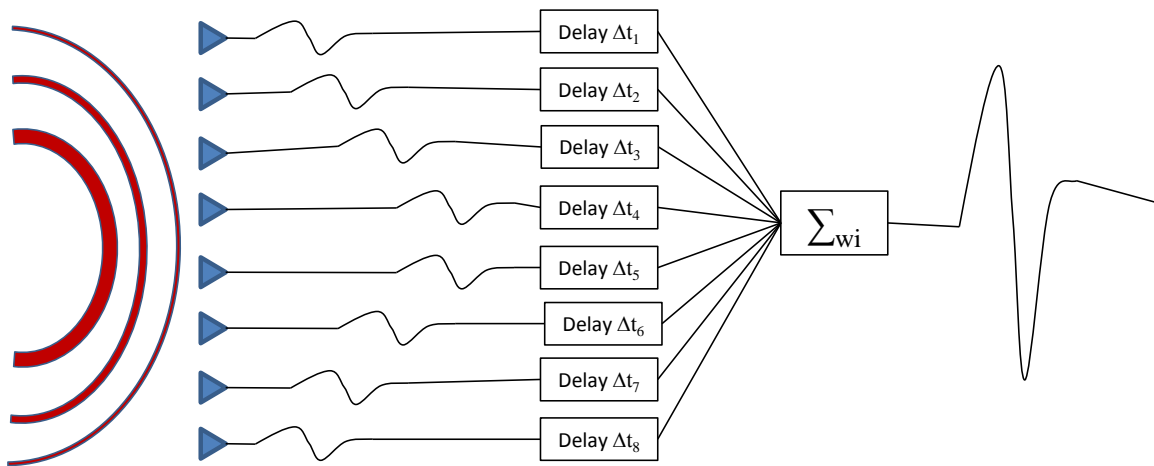


Figure 4. Representation of delay and sum beamforming. After [16].

Computing array amplitude and phase weights according to the Dolph-Chebyshev method allows the selection of a specific uniform side lobe suppression level [13]. This setup works well with small and large arrays and makes possible a generic method for incoming acoustic signal direction detection. The method for computing Dolph-Chebyshev amplitude weights comes from [13] and is an approximation. This approximation was found to hold for up to 42 hydrophone elements. For the actual Simulink model, a different approximation for calculating the Dolph-Chebyshev amplitude weights was used that continues to perform past 100 hydrophone elements. The TLA model uses the better approximation and is an openly available Matlab function that can be downloaded from the Mathworks website in a package called “Electromagnetic Waves & Antennas Toolbox” [17]. The better approximation mirrored the approximation given by [13] until 42 elements and performs as one would expect the Dolph-Chebyshev method to for additional elements. The approximation described by [13], outlined below, provides insight to how the amplitude weights are generated.

Dolph-Chebyshev amplitude weights are determined approximately by first defining the amount of side lobe suppression desired R in dB. This is used to find a coefficient x_0 which is used later in the derivation of amplitude weights. The value of the suppression level ratio r is defined as

$$r = 10^{\frac{R}{20}}. \quad (1)$$

In order to determine x_0 it is necessary to solve a m th degree Chebyshev polynomial which is defined according to [13]

$$T_m(x) = \begin{cases} \cos[m \cos^{-1}(x)], & |x| < 1 \\ \cosh[m \cosh^{-1}(x)], & x \geq 1 \\ (-1)^m \cosh[m \cosh^{-1}(-x)], & x \leq -1 \end{cases}. \quad (2)$$

The coefficient x_0 can be found by solving

$$T_i(x_0) = r \quad (3)$$

where N is the number of hydrophone elements and the order of the Chebyshev polynomial is defined by $i = N - 1$. Solving Equation (3), we get

$$x_0 = \cosh \left[\frac{1}{i} \cosh^{-1}(r) \right], \quad r > 1. \quad (4)$$

Now that x_0 has been determined, the amplitude weights can be found. The set of amplitude weights for an even number of hydrophones elements is found from

$$a_n(f) = \sum_{j=n}^{N/2} \left[(-1)^{\left(\frac{N}{2}-j\right)} \right] \frac{N-1}{N+2(j-1)} \binom{\left(\frac{N}{2}\right)+j-1}{2j-1} \binom{2j-1}{j-n} x_0^{2j-1}, \quad n = 1, 2, \dots, N/2, \quad (5)$$

where the binomial coefficient used is defined as

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}. \quad (6)$$

For the odd element case, binomial coefficients are used again and a new variable N' is defined as

$$N' = \frac{N-1}{2}. \quad (7)$$

The set of amplitude weights for an odd number of hydrophone elements is found from

$$a_n(f) = \sum_{j=n}^{N'} \left[(-1)^{(N'-j)} \right] \frac{N'}{N'+j} \binom{N'+j}{2j} \binom{2j}{j-n} x_0^{2j}, \quad n = 0, 1, \dots, N'. \quad (8)$$

Now that amplitude weights have been chosen so that the main lobe exceeds the side lobes by the desired level, the next step is to determine where to steer individual beams for bearing determination. An example of a 10 element array with a main lobe

steered to 90 degrees and a side lobe suppression level of -20 dB is shown in Figure 5. Clearly, the main lobe is 20 dB above the various side lobes, all of which are suppressed to the same level.

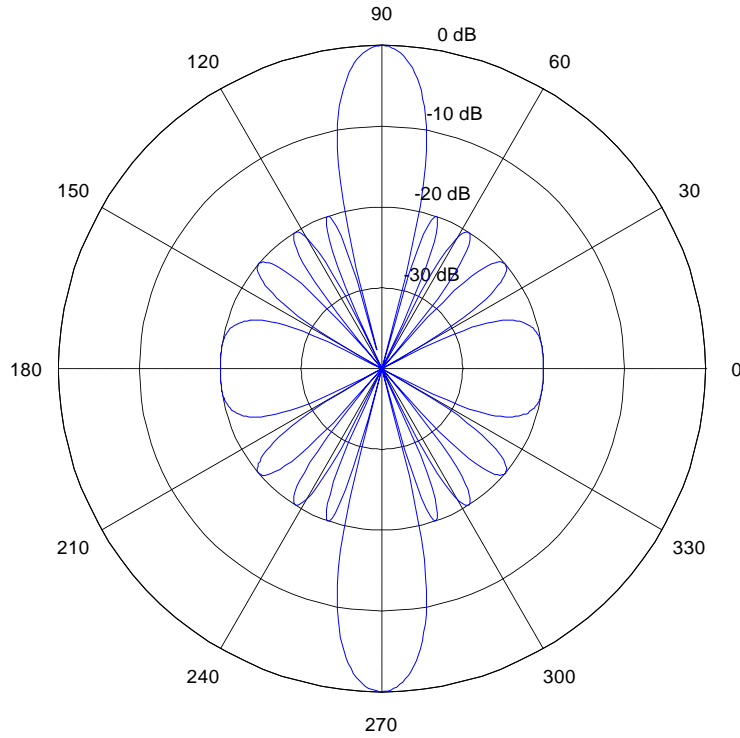


Figure 5. Main lobe steered to 90 degrees formed using Dolph-Chebyshev optimization method.

3. Determining Signal Bearing

Incoming acoustic signal bearing is determined by comparing the power of multiple steered main lobes to each other. To accomplish this, one must first be able to steer a main lobe. The next step is to determine how many main lobes are necessary or possible to steer and process. The last step in the bearing determination process is to use an algorithm to determine a bearing from the power signals from the various beams processed.

a. Beam Steering

Beam steering can be accomplished by either time delaying the various input signals to look down the desired bearing or frequency shifting the signal in the Laplace domain. Because Simulink with variable size arrays was used in this thesis, the time delay method was used. The time delay for hydrophone element n used to generate a desired steering angle ψ' is determined by

$$t'_n = \frac{\cos(\psi' nd)}{SS}, \quad n = \begin{cases} -N', \dots, 0, \dots, N' & (\text{for odd } N) \\ -N, \dots, -1, 1, \dots, N & (\text{for even } N) \end{cases}, \quad (9)$$

where ψ' is the desired angle to steer the main lobe, and SS is the speed of sound for the conducting medium. The assumption here is that the elements are equally spaced and there is only steering in the X-Y plane.

b. Beam Spacing

The main beams were spread to overlap at the 3 dB down points in order to make the bearing determination from incoming acoustic signals as simple as possible. This is accomplished by first steering a beam to broadside and determining its beamwidth, which is defined by the 3 dB down points. Once the beamwidth is calculated, two new beams are steered using the 3 dB down bearing on either side of the broadside lobe. This process continues until the next beam 3 dB down point occurs more than 90 degrees from broadside. Beamwidth $\Delta\psi$ of a main lobe formed using the Dolph-Chebyshev method is given by [13]

$$\Delta\psi = 180^\circ - 2 \cos^{-1} \left[\frac{\lambda}{\pi d} \cos^{-1}(x_+) \right], \quad (10)$$

where x_+ is the value of x that corresponds to the 3 dB down point ψ_+ and is defined by

$$x_+ = \frac{1}{x_0} \cosh \left[\frac{1}{i} \cosh^{-1} \left(\frac{\sqrt{2}r}{2} \right) \right]. \quad (11)$$

The equation for x_0 is given in Equation (4), and r is defined by Equation (1). The value for i is the same as before and is defined as $i = N - 1$.

c. Bearing Determination

At this stage the proper amplitude weighting for main lobe shape is determined, and a set of these main lobes, referred to as beams, has been spread out overlapping at 3 dB down points from broadside. After much time spent adjusting the beam spacing and processing them for bearing, it is observed that, in our experience, the best process is to remove two beams from the extremes on either side of the broadside lobe and replace them with a beam centered on 0° and 180° , respectively. For almost all cases observed, this removed beams that had grating lobes or overlapped enough to make it difficult to determine a direction finding algorithm. Using this same approach led to the development of an algorithm that can be generalized and used to determine bearing at the array endfires. The term endfire is used to describe the direction along the length of the linear array. It is a tedious process to determine how to discern bearings near the endfire angles without using this process.

For the interior beams, which are spread by 3 dB down points, the bearing determination scheme is quite simple. The maximum power beam for an output time step, which is generally greater than the simulation time step, is determined, and the power levels of the two adjacent beams are compared to find the second highest power beam. The beams are numbered $1:m$. The ratio γ of the maximum power beam to next most powerful beam is given by

$$\gamma = \frac{P_{\psi_{\max}}}{P_{\psi_{2nd}}} = \begin{cases} \sqrt{2} & , \beta = \psi'_{\max} \\ 1 & , \beta = \psi'_{\max} \pm \frac{|\psi'_{\max} - \psi'_{2nd}|}{2} = \psi'_{\max} \pm \alpha_m, \\ 1 < \gamma < \sqrt{2} & , \text{Otherwise} \end{cases} \quad (12)$$

where α is defined as the bearing between two beams. The bearing when γ is not equal to one of the special cases is equal to

$$\beta = \psi'_{\max} \pm \alpha_m \left[1 - \left(\frac{\gamma - 1}{\sqrt{2} - 1} \right) \right]. \quad (13)$$

The sign of the ratio term depends on which side of the most powerful beam the second most powerful beam is located. The layout for a generic seven element array with nine beams is shown in Figure 6.

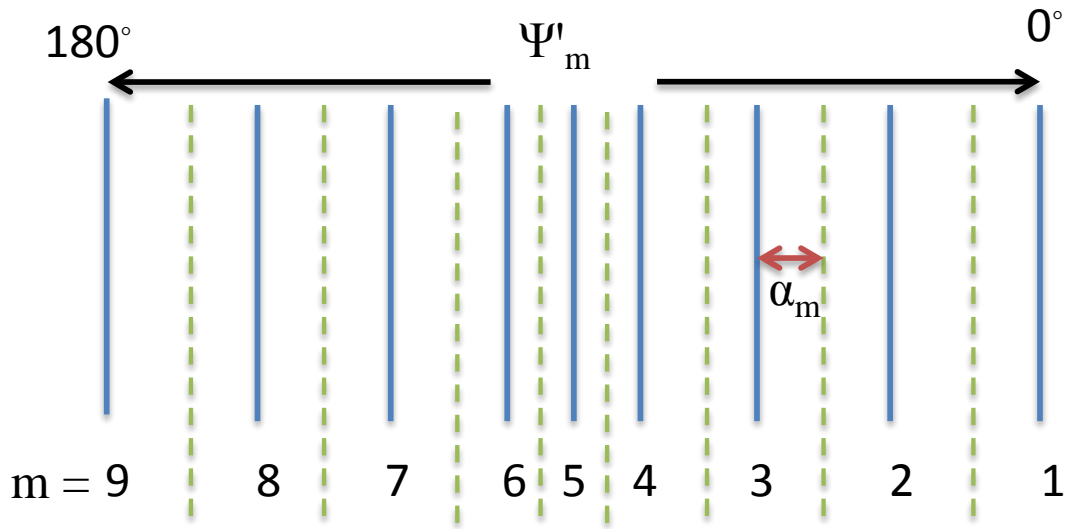


Figure 6. Graphical display of beams and angles for a generic seven-element array.

The spread between the first and second, as well as the spread from the last and second to last beam, does not occur at the 3 dB down point. In order to best determine how to discern bearings for this section, the following process is very helpful and may save some time if it is desired to discern bearings near endfires for a generic setup. It should be noted that this has been found by trial-and-error and may not be the best method. With that said, it does seem to work well for all the cases that have been run using this method.

The first step is to run a TLA model simulation and sweep the target bearing from one endfire to the other. Next, for the front endfire, plot the ratio of the

received power γ for the first to second beams. There are three coefficients that are used. The first parameter A is set to value of the bearing β as a ratio of α that corresponds to the bearing where $\gamma = 1$. The second coefficient B is set to the maximum value of γ observed for a near zero bearing. The third coefficient C is set to the value of γ where the bearing is equal to ψ'_2 . The plot of γ over time is displayed as an example of the process used to generate the coefficient values A , B and C in Figure 7.

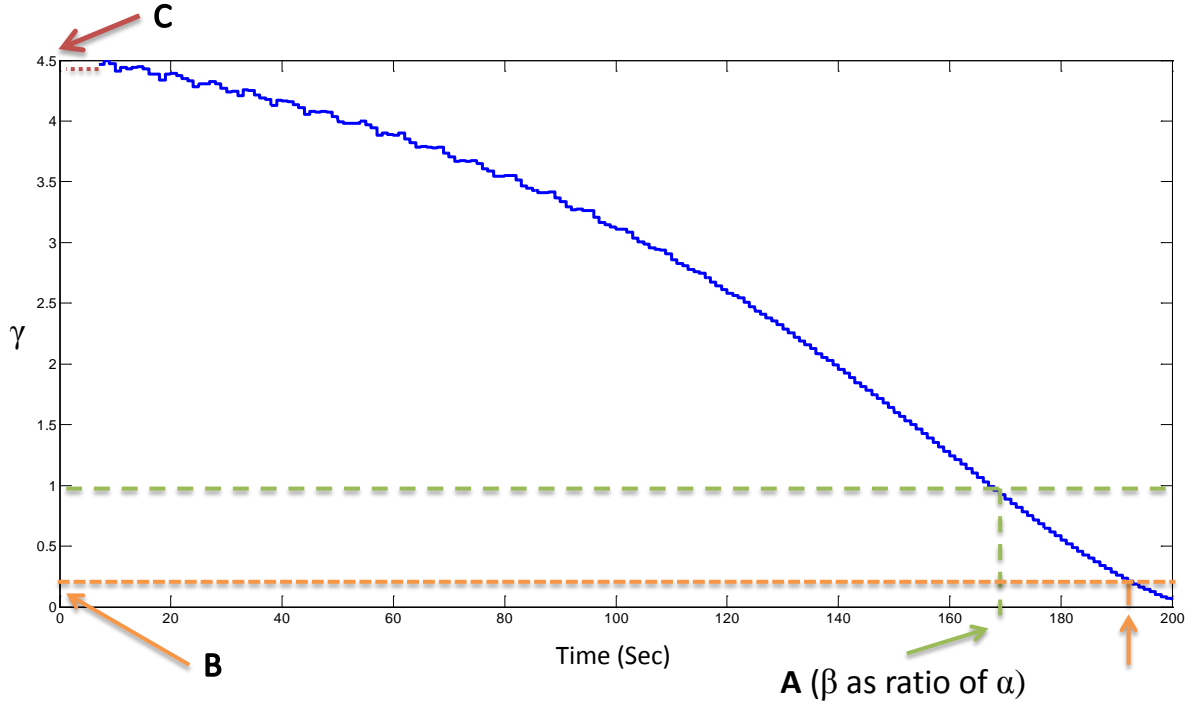


Figure 7. Plot of beam one to beam two power ratios vs. time used to generate coefficients for endfire bearing determination.

With the above derived coefficients, bearing at the forward endfire is

$$\beta = \begin{cases} A\alpha - A\alpha \left(\frac{\gamma - 1}{B - 1} \right) & , \psi'_{\max} = 1 \\ A\alpha + 0.3\alpha \left[1 - \max \left(0, \frac{\gamma - C}{1 - C} \right) \right] & , \psi'_{\max} = 2 \text{ \& } P_{\psi_1} > P_{\psi_3} \end{cases} \quad (14)$$

A similar process is used to determine how to discern the bearings at the aft endfire. Application of this process results in an approximation of the bearing in the endfires that normally do not produce a step jump between the various beam regions. The ratio of endfire beam to first interior beam powers are subject to many outside forces like range and emitted source level. This means that using this process gives some indication of bearing to target when the target is in the endfire regions, but depending on the many variables, it will generally be biased. Any good submariner will likely tell you not to trust any data coming from an endfire. Using this technique provides indications of bearing in the endfire regions that generally should not be trusted because they are more than likely biased.

D. TLA MODEL PERFORMANCE

A simplified depiction of how to implement the dynamic TLA model in Simulink is shown in Figure 8. Allowing the model to retain the ability to handle differing sizes of arrays is a both an advantageous feature and limiting feature. Differing sized arrays require the output from some model components to be variable in size. This is not something that the Embedded Matlab blocks can handle [18], which results in longer simulation times. It is necessary to code components that need variable output sizes as user defined functions. User defined functions do not get the support of the features in Simulink, which greatly speed up the simulation times when the Embedded Matlab blocks are used. This resulted in very long simulation times. The seven-element case, presented later, took almost an hour to run an hour of data. The 100-element case took two full days to run less than an hour of data.

A small array that could be deployed from a UUV or USV, a medium length array and a longer array that would be representative of an array that could be deployed from a ship or submarine were modeled. The base case for geometry was chosen that sweeps the target from the forward endfire to the aft endfire. The array parameters, for the different cases and geometrical setup of the simulation are shown in Table 1.

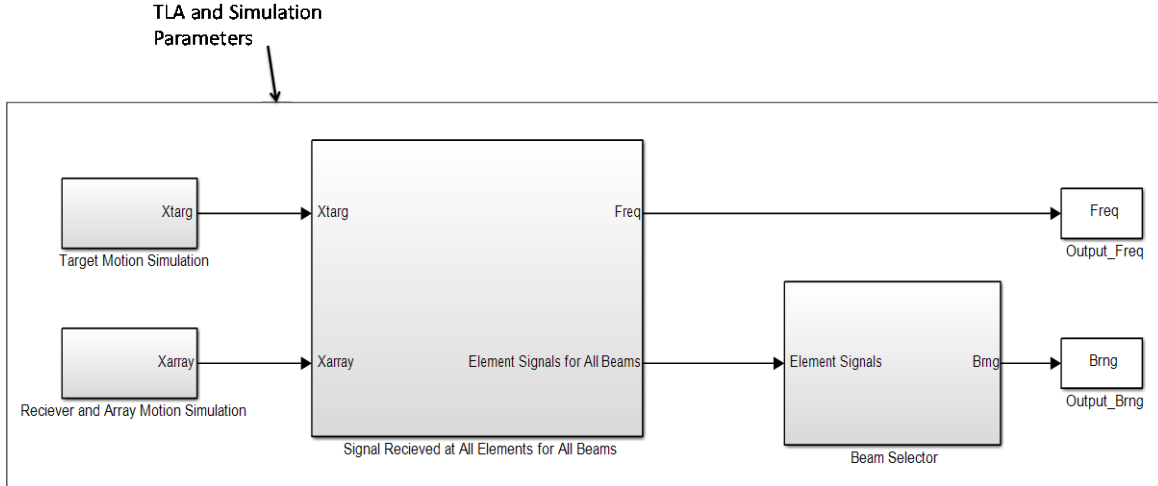


Figure 8. A general depiction of how to implement a dynamic TLA model.

Table 1. Parameters used for TLA test cases.

| TLA Case Summary | | | |
|---------------------------------|---------|---------|---------|
| | Case A | Case B | Case C |
| N | 7 | 21 | 100 |
| fcrit (Hz) | 50 | 50 | 50 |
| λ (m) | 30 | 30 | 30 |
| d (λ) | 0.4 | 0.3 | 0.2 |
| d (m) | 12 | 9 | 6 |
| Length (m) | 72 | 180 | 594 |
| fmin (Hz) | 20.8 | 8.3 | 2.5 |
| fmax (Hz) | 62.5 | 83.3 | 125 |
| Array Pos. (NM) | (0,0) | (0,0) | (0,0) |
| Array Vel. (kn) | (20,0) | (20,0) | (20,0) |
| Target Pos. (NM) | (20,5) | (20,5) | (20,5) |
| Target Vel. (kn) | (-20,0) | (-20,0) | (-20,0) |
| Sim Length (Sec) | 3600 | 3600 | 2700 |
| Sim Step (Sec) | 0.005 | 0.005 | 0.005 |
| Output Step (Sec) | 0.5 | 0.5 | 0.5 |
| R (dB) | -30 | -30 | -30 |
| A | 1.4 | 1.4 | 1.4 |
| B | 0.5 | 0.5 | 0.5 |
| C | 3.5 | 3.5 | 3.5 |

For the seven-element array, the far-field beam pattern is shown in Figure 9. There are no grating lobes present and there is a significant bearing spread, greater than fifty degrees, between the endfire beams and the next beam. The beams are almost all

suppressed to the desired -30 dB level. The first interior beam is the exception, but since the non-desired portions of the lobe are suppressed by more than -10 dB, it presents no problem. The seven-element case has nine very large lobes that are wide.

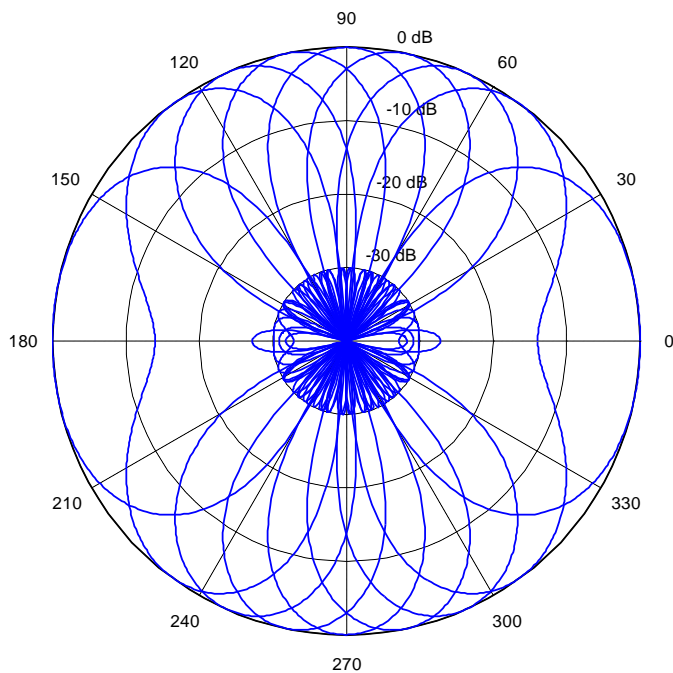


Figure 9. Far-field beam pattern for Case A towed linear array.

As time progresses in the model, the targets sweeps from the forward endfire at a long range through a 5 NM closest point of approach and then to the aft endfire at longer range again. The output power of each beam, sampled every half of a second, is shown in Figure 10. The beam powers increase as the target is down the angle that the beam is steered, as expected. The power also increases as the target closes in range and then falls again as the range increases. It is also obvious that when the target is not close to the angle of the beam that the signal power level coming from that beam is comparatively very low.

The true and measured bearing from the Case A array are shown in Figure 11. For the interior part of the array, which contains most of the beams, the TLA

performance is very good. At the endfires, the relative bearing motion is captured, but clearly the coefficients for the endfire are not optimized for the range of this scenario.

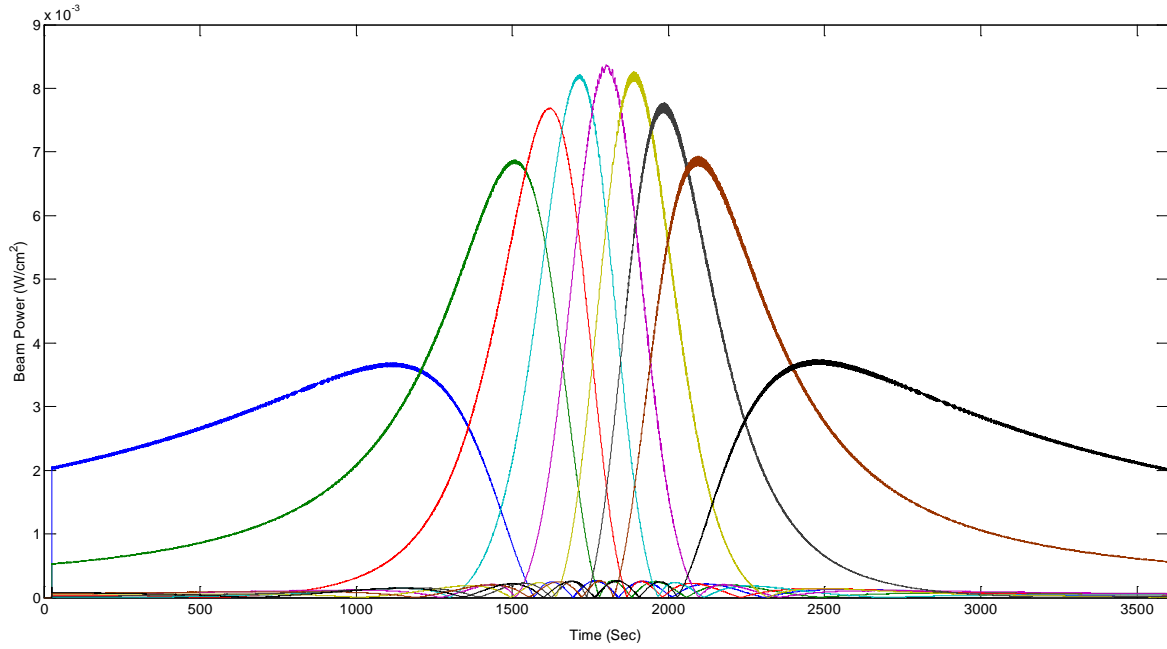


Figure 10. Beam power signals for Case A simulation run.

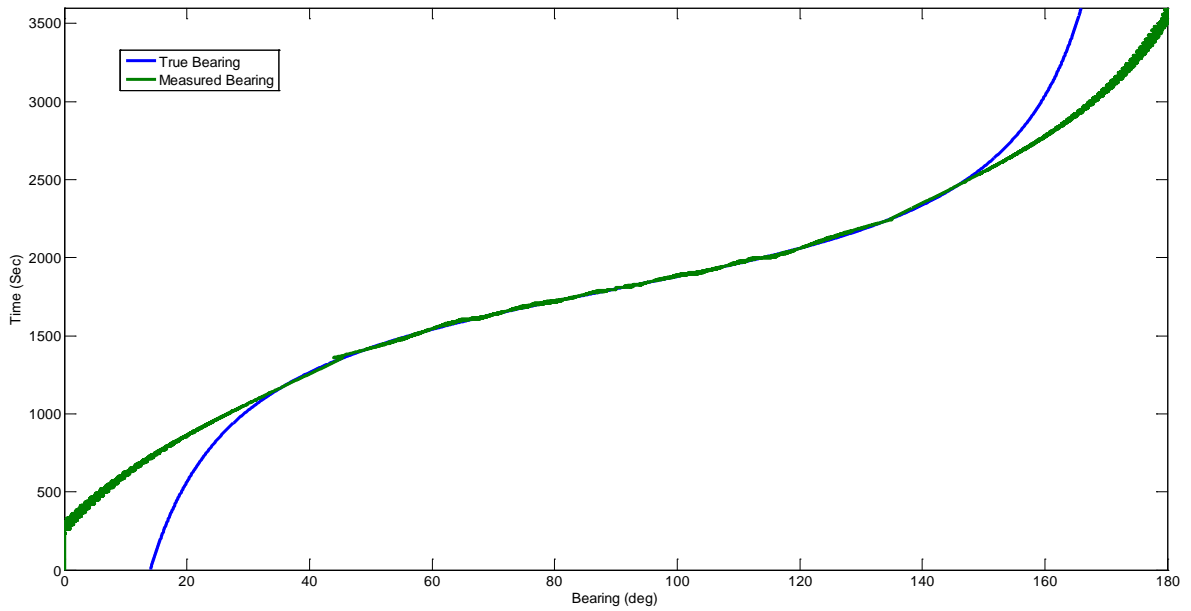


Figure 11. Case A true and measured bearing over time.

The performance of the seven-element TLA is good for the interior portions of the array, where the beams are concentrated, and worse at the endfires. The first interior beam is situated at 53 degrees, which means that there is a large bearing spread that must be discerned without good information. The run time of the simulation with seven elements was sufficiently fast to gather more data than the time required for simulation.

The next case, B, details the performance of a 21-element TLA. The far-field beam pattern for a 21-element TLA, formed using Dolph-Chebyshev Optimization, is shown in Figure 12. It is easy to see that the beams are narrower and that there is greater beam coverage when approaching the endfires than with the seven-element case.

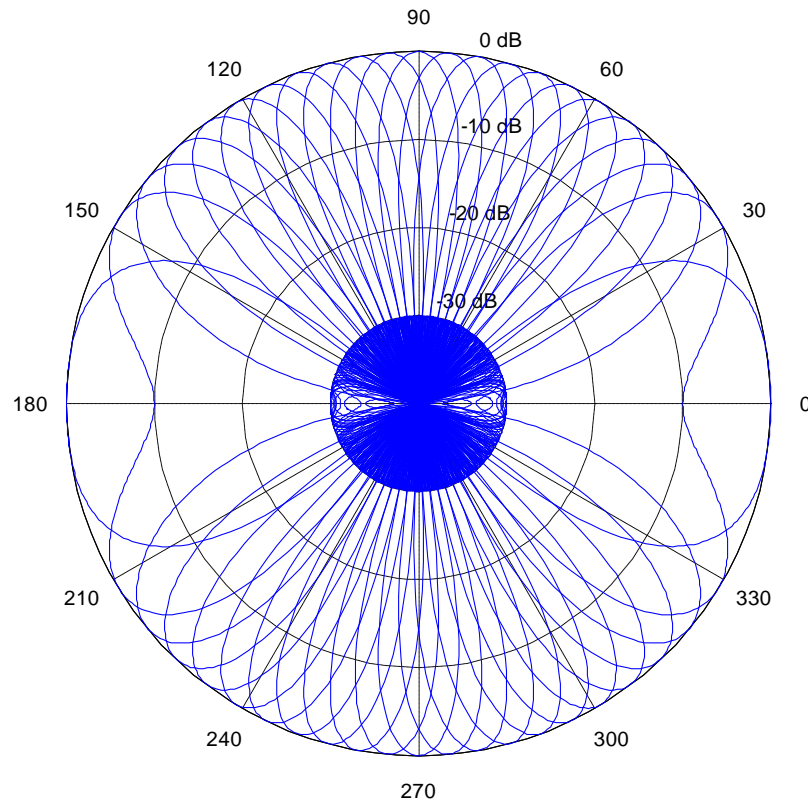


Figure 12. Far-field beam pattern for Case B towed linear array.

The output beam power for Case B is plotted in Figure 12. The true and measured bearings over time are plotted in Figure 13. Overall, the larger number of

beams results in less bearing error over time. The maximum beam power again shifts from forward beams to aft as the scenario progresses. Clearly, this case shows good TLA behavior. The run time is significantly longer, taking many hours to complete.

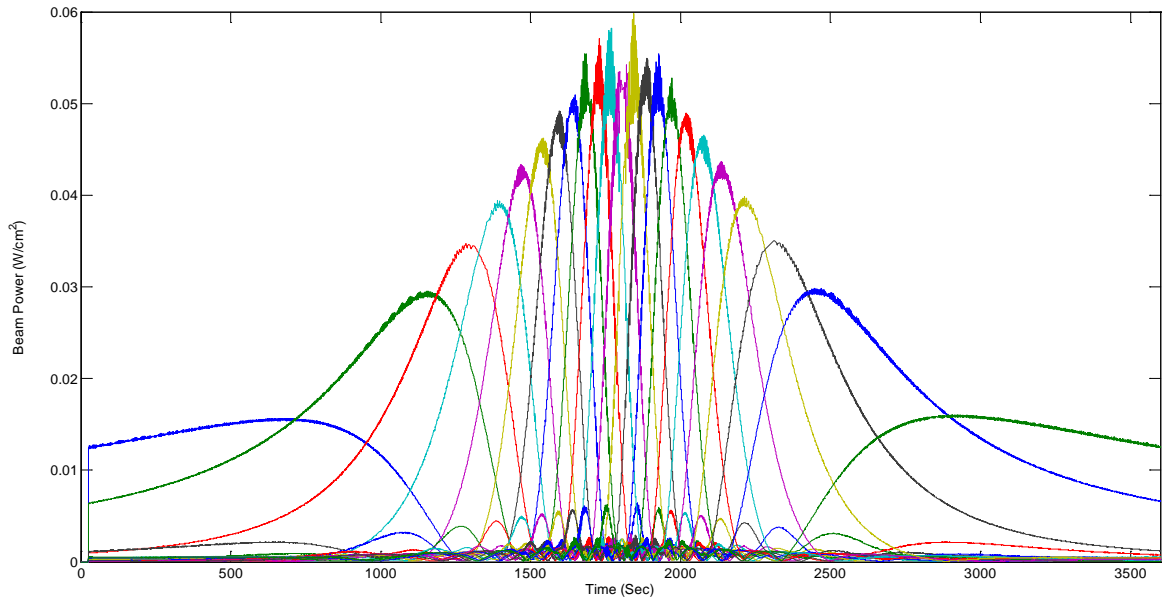


Figure 13. Beam power signals for Case B simulation run.

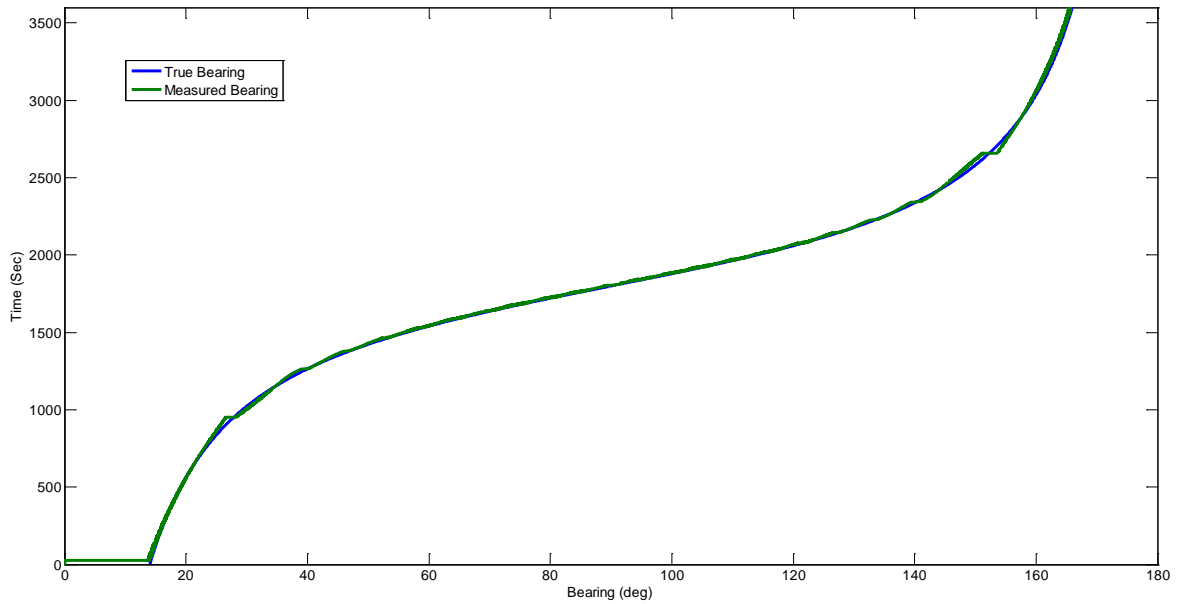


Figure 14. Case B true and measured bearing over time.

The initial large change in measured bearing is due to the way that the array is modeled. The array positions are dependent upon the past positions of the towing platform; therefore, as more time progresses more of the array is formed. The longer the array is, the longer the initialization takes, and the longer the initial data errors occur. This will be readily apparent in the 100-element case.

Case C describes the output from modeling a 100-element array. The far-field beam pattern for Case C is shown in Figure 15. Clearly, there are more beams, and they are much narrower, providing a larger degree of directivity. The beams also again cover more ground as they spread toward the endfires.

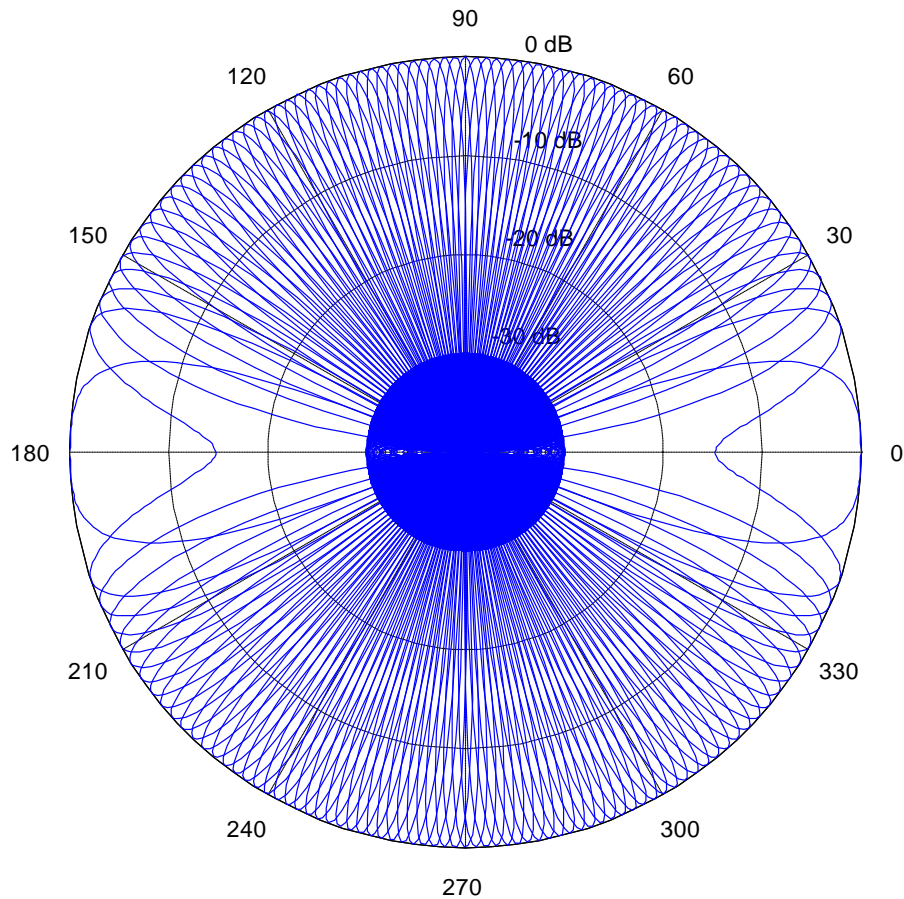


Figure 15. Far-field beam pattern for Case C towed linear array.

The output beam power for Case C is plotted in Figure 16. The true and measured bearings over time are plotted in Figure 17. The maximum beam power again shifts from forward beams to aft as the scenario progresses.

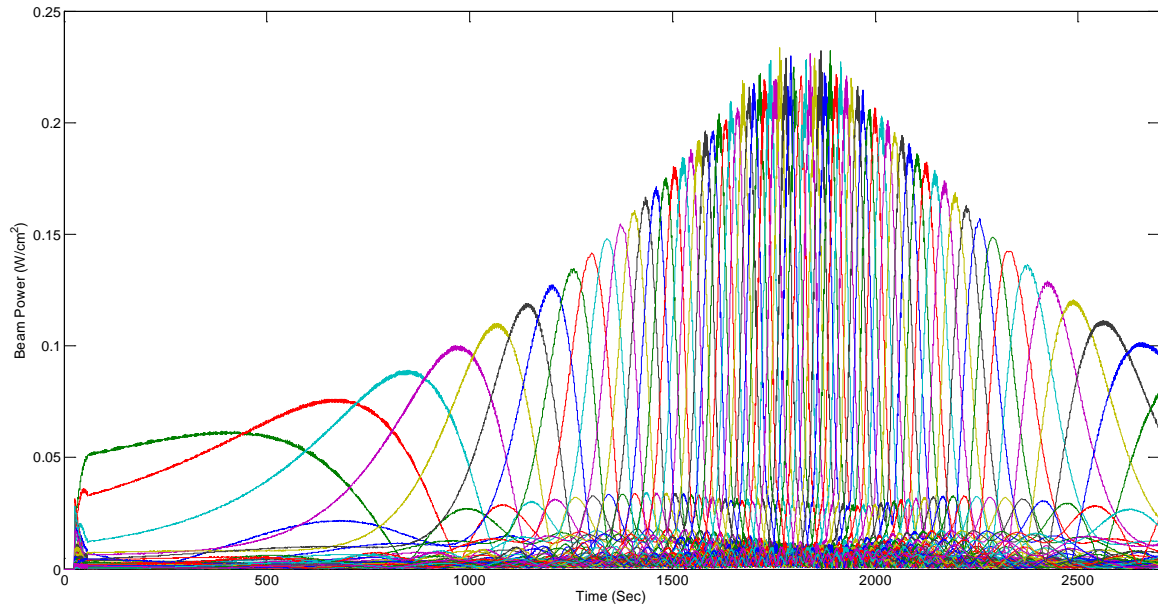


Figure 16. Beam power signals for Case C simulation run.

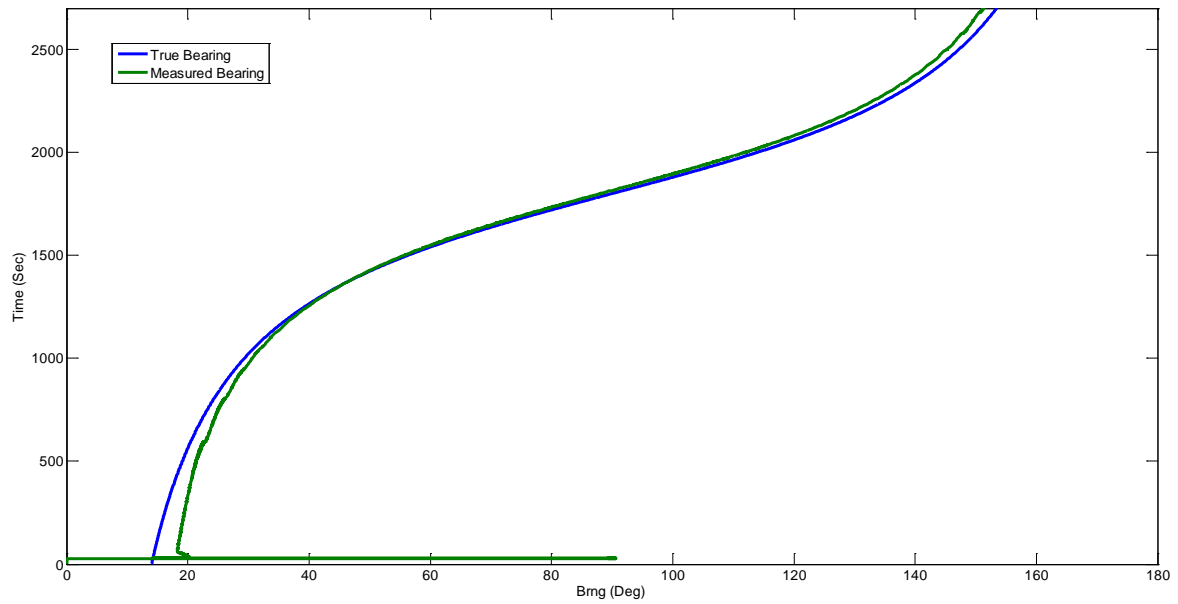


Figure 17. Case C true and measured bearing over time.

For all cases, the bearing tracked well during the interior portions of the array, and there were no large step changes in bearing, with the exception of data initialization. The simulation time clearly is too long for most applications, with the larger array case taking in excess of two full days to complete. It is important to note that the output measured bearing was only plotted on the same side as truth. In reality, there would be no way on one leg to determine which side of the array the signal was coming from without more data. The receiver never maneuvered in the previous cases. It is necessary to see what kind of behavior can be expected out of the array during a turn. A seven-element case is run again with the parameters depicted in Table 2.

Table 2. Maneuvering receiver simulation Case A1 parameter setup.

| Case A1 | |
|---------------------------------|---------|
| N | 7 |
| fcrit (Hz) | 50 |
| λ (m) | 30 |
| d (λ) | 0.4 |
| d (m) | 12 |
| Length (m) | 72 |
| fmin (Hz) | 20.8 |
| fmax (Hz) | 62.5 |
| Array Pos. (NM) | (0,0) |
| Array Vel. (kn) | (15,0) |
| Target Pos. (NM) | (20,5) |
| Target Vel. (kn) | (-20,0) |
| Sim Length (Sec) | 3600 |
| Sim Step (Sec) | 0.005 |
| Output Step (Sec) | 0.5 |
| R (dB) | -30 |
| A | 1.4 |
| B | 0.5 |
| C | 3.5 |
| Turn | Port 45 |
| Turn Time (Sec) | 1800 |

The output beam power for Case A1 is plotted in Figure 18. The true and measured bearings over time are plotted in Figure 19. The maximum beam power shifts from forward beams to aft as the scenario progresses until the turn. At the turn the power signals shift quickly and then resume the shift to aft beams, becoming the ones with more power.

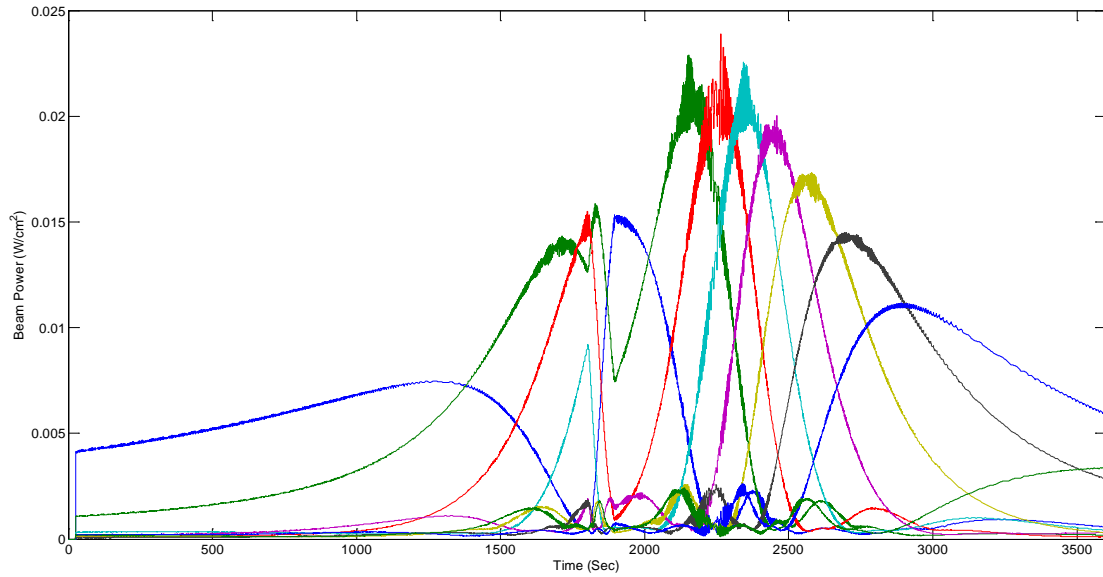


Figure 18. Beam power signals for Case A1.

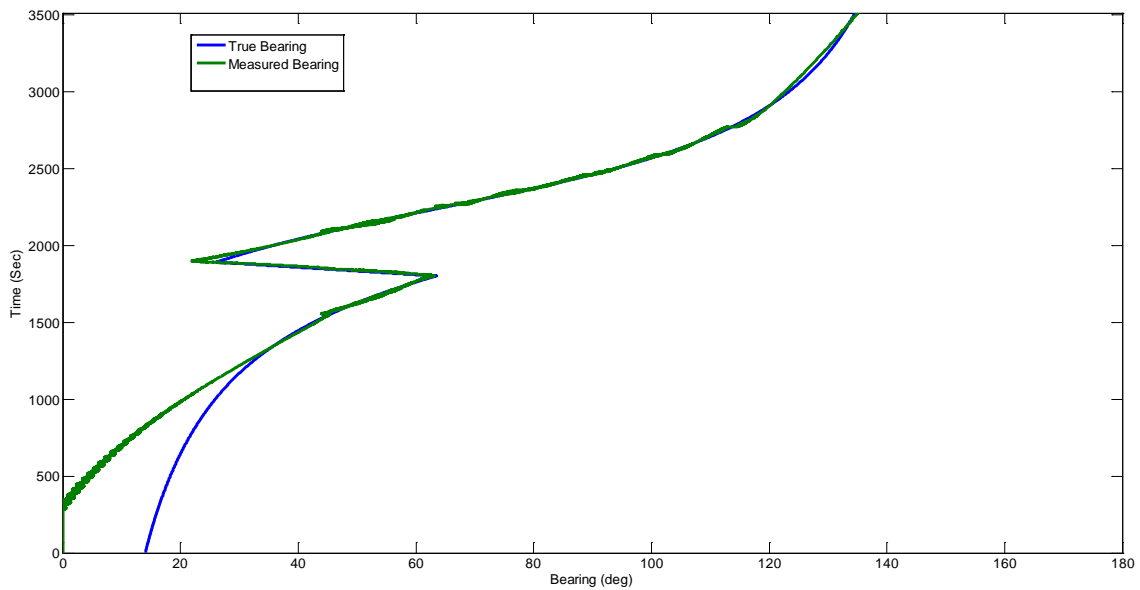


Figure 19. True and measured bearing over time for Case A1.

In this simulation the bearing actually followed pretty well through the turn. There is a bit of bearing overshoot during and after the turn. The ability to track through the turn here is unrealistic because there is no oscillation of the array as it turns, and it returns to its straight shape quickly in the simulation. The turn rate used, one half a degree per second, is greater than a submerged contact could manage. If a contact could turn this fast, it would produce a large area of turbulence in the water as the rudder would be shifting quickly.

The performance of the TLA model is better than one could expect in real life. The array is able to turn and stabilize quicker than a real deployed array could manage. The model was made in a manner that makes it quite easy to insert a longer delay for straightening out, as well as a perturbation that would correspond to the oscillation that occurs due to the forces of the turn.

E. CHAPTER SUMMARY

In this chapter, it was shown how to go about generating a dynamic TLA model that produces output bearings that match well with truth and have no large jumps when transitioning between different beam areas. The limitations of receiver platform turning were described. The long simulation run length was described including why it occurs. In the next chapter how to generate a target state estimate using the output from the TLA is detailed. The generation of target state estimation is accomplished by using a BO-EKF and DB-EKF.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TARGET STATE ESTIMATION

A. INTRODUCTION

An introduction to a TLA and how to model it were presented in Chapter II. Now that it has been shown how to take a linear hydrophone array and process incoming signals to output bearing, frequency and SNR data, it is necessary to see how one could track a target with this data in hand. The Extended Kalman Filter (EKF) is the focus of this chapter. The BO-EKF is discussed as well as the DB-EKF. The merits of both filters is detailed and the limitations and problems with the use of these filters for tracking is discussed.

B. CURRENT TLA TARGET STATE ESTIMATION TECHNIQUES

As discussed in Chapter II, the most common type of military TLA used is a single passive TLA with omnidirectional elements having little or no information on the state of the array. Therefore, the most common method for estimating a maneuvering target's state is presented using this type of array.

The U.S. Navy uses a variety of tracking techniques ranging from a hand drawn Geographical Plot to parallel processing of large state possibilities and comparing resulting solutions in a least squares manner. While these techniques work, there is still the problem of the level of confidence that should be placed into the estimated solution. This is not a new problem and was summarized succinctly over three decades ago by [7]: “The CO lacks an adequate assessment of the degree of confidence he should place in a ranging solution. He may be unable, therefore, to make a well-founded choice between continued data collection and analysis versus immediate attack.” The focus here is tracking. The same problem concerning the assessment of the estimation state uncertainty still makes keeping the tracking platform safe while maintaining track of the target platform a hard situation for all involved.

Extended Kalman filters work well at tracking maneuvering targets, as demonstrated by their prolific use in the radar tracking arena. One advantage of EKFs is that the status of the uncertainty in the state estimate is tracked over time while

generating a state solution for the target. This means that the status of the uncertainty of the estimate can be displayed to operators as the tracking problem progresses in real time. There are problems associated with EKF's and their use with tracking in a setting like submerged maneuvering tracking. The problems with EKF's include biasing and no guarantee of convergence. These problems can be overcome using various techniques that often prohibit the filters from running in real time. Since a real time solution to the tracking problem is desired that presents the platform operator with a state covariance estimate, the EKF is explored.

C. TARGET STATE ESTIMATION VIA EXTENDED KALMAN FILTERS

1. EKF Background

A Kalman Filter (KF) recursively estimates the state of a linear system from measurements taken with zero mean Gaussian noise in an optimal minimum mean square sense [19]. An EKF extends the functionality of the KF to the non-linear realm. The non-linear dynamics of the system are linearized using a series expansion and results in a linear, minimum mean-square error estimation [20]. One benefit of using a KF or EKF is that one has access to the covariance matrix of the least squares estimate, which provides a direct indication of how good the solution is perceived to be. Using the state estimate along with the covariance matrix, one can plot not only the best estimate of the target state but the uncertainty of the estimate. A more detailed description of the EKF and KF can be found in [19, 20, 21] or any good text on state estimation.

2. Bearing-Only Tracking Using BO-EKF

a. Overview

The position and velocity components of the receiver platform are known. It is necessary to estimate the state of the target platform. The target state vector is given by

$$\vec{x}_t = \begin{bmatrix} x_t \\ v_{xt} \\ y_t \\ v_{yt} \end{bmatrix} = \begin{bmatrix} x - position \\ x - velocity \\ y - position \\ y - velocity \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (15)$$

The receiver state, known at all times, is defined by

$$\vec{x}_r = \begin{bmatrix} x_r \\ v_{xr} \\ y_r \\ v_{yr} \end{bmatrix}. \quad (16)$$

The EKF is a recursive algorithm. The current time step is defined as k and the next time step is $k+1$. In discrete time the state vector changes with time according to

$$\vec{x}(k+1) = F(k)\vec{x}(k) + \eta(k) = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \vec{x}(k) + \eta(k) \quad (17)$$

where the value of the time step is defined by Δ , $\eta(k)$ represents zero mean white Gaussian process noise, and $F(k)$ is the state transition matrix. The state transition matrix in this case assumes that the target is normally moving in a straight line. Clearly, the target does not always move in a straight line. Non-straight line motion is taken into account with the covariance matrix of the process noise. This means that the filter is expecting the target to move in a straight line and any deviation from straight line motion is modeled as random zero mean process noise. The process noise covariance matrix is defined as

$$E[\eta(k)\eta(k)'] = Q(k), \quad (18)$$

where

$$Q(k) = \begin{bmatrix} \frac{q^2 \Delta^3}{3} & \frac{q^2 \Delta^2}{2} & 0 & 0 \\ \frac{q^2 \Delta^2}{2} & q^2 \Delta & 0 & 0 \\ 0 & 0 & \frac{q^2 \Delta^3}{3} & \frac{q^2 \Delta^2}{2} \\ 0 & 0 & \frac{q^2 \Delta^2}{2} & q^2 \Delta \end{bmatrix} = q^2 \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta \end{bmatrix}. \quad (19)$$

Here q^2 is a coefficient representing the unknown acceleration in the system. This means that adjusting the value of q^2 impacts how much non-linear acceleration the model can handle. In [22] it is shown that the best value for q^2 is generated by determining how much acceleration is expected.

The only measurement that is used to estimate the target state is the relative bearing β to the target. The non-linear measurement equation which describes the value of the measurement $z(k)$ from the state values is

$$z(k) = h(\vec{x}_t, \vec{x}_r, k) + \omega(k) = \arctan\left(\frac{y_t(k) - y_r(k)}{x_t(k) - x_r(k)}\right) - \arctan\left(\frac{V_{ry}}{V_{rx}}\right) + \omega(k) = \beta(\vec{x}_t, \vec{x}_r, k) - Hdg + \omega(k), \quad (20)$$

where Hdg is the receiver heading and ω is the zero mean Gaussian measurement noise, which has a covariance equal to

$$R(k) = [\sigma_\beta^2]. \quad (21)$$

The term σ_β^2 represents the bearing measurement variance. The measurement estimate for time k given $k-1$ measurements is given by

$$\hat{z}(k | k-1) = h(\vec{x}_t, \vec{x}_r, k | k-1) = \beta(\vec{x}_t, \vec{x}_r, k | k-1) = \arctan\left(\frac{\hat{y}_t(k | k-1) - y_r(k)}{\hat{x}_t(k | k-1) - x_r(k)}\right) - Hdg. \quad (22)$$

The gradient of $h(\vec{x}_t, \vec{x}_r, k)$ with respect to \vec{x}_t is defined as $H(\vec{x}_t, \vec{x}_r, k)$ and has the form

$$H(\vec{x}_t, \vec{x}_r, k) = \begin{bmatrix} \frac{y_r - y_t}{\sqrt{(x_t - x_r)^2 + (y_t - y_r)^2}} & 0 & \frac{x_t - x_r}{\sqrt{(x_t - x_r)^2 + (y_t - y_r)^2}} & 0 \end{bmatrix}_{\vec{x}_t = \vec{x}_t^{\oplus}(k|k-1)} . \quad (23)$$

To start the state estimation process, one needs an initial estimate of the target state \hat{x}_{int} and initial predicted state covariance $P(0|0)$. The values for $P(0|0)$ are large if the values of \hat{x}_{int} are uncertain and small if you trust \hat{x}_{int} . Generally, the values of $P(0|0)$ must be derived depending on the estimation technique for \hat{x}_{int} .

With initial predicted state values and initial predicted state covariance in hand, the next step in the recursive state estimation process is to update the state prediction covariance

$$P(k+1|k) = F(k)P(k|k)F(k)' + Q(k) . \quad (24)$$

From here the next step is to compute the Kalman Gain $W(k)$ from

$$W(k+1) = P(k+1|k)H(k+1)'[H(k+1)P(k+1)H(k+1)' + R(k+1)]^{-1} . \quad (25)$$

The next step is to update the state estimate and predicted state covariance

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)[z(k+1) - \hat{z}(k+1|k)] , \quad (26)$$

and

$$P(k|k) = (I - W(k)H(k))P(k|k-1)(I - W(k)H(k))' + W(k)R(k)W(k)' , \quad (27)$$

respectively. Here the identity matrix of the same size as the state covariance matrix is used. This is usually the time that the results for \hat{x} are posted. Next, the state is estimated for the next time step from

$$\hat{x}(k+1|k) = F(k)\hat{x}(k) . \quad (28)$$

b. *BO-EKF Model*

In general, implementing this EKF in Simulink is very easy. The recursive algorithm described in the previous section is coded in an embedded Matlab function block. It is important to note that in order for the embedded Matlab function block to compile, a compatible C compiler must be installed on the computer in use. It is worth the effort to program the filter in the embedded Matlab function block, as it can handle multiple inputs and outputs and runs faster than using the “User defined function” block. What in general is necessary to implement a BO-EKF in Simulink is shown in Figure 20.

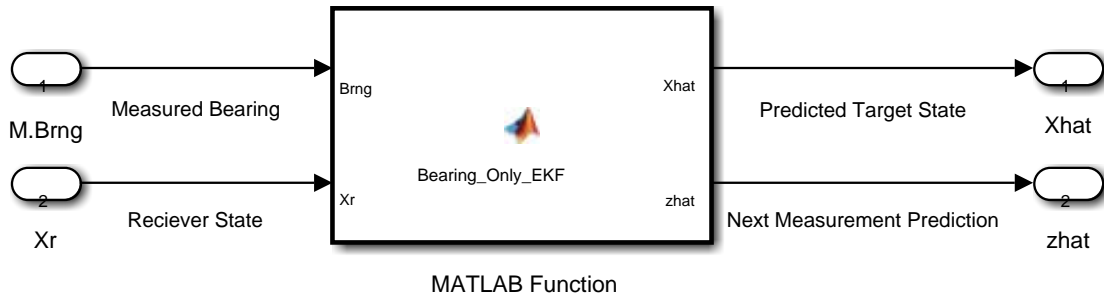


Figure 20. BO-EKF implementation in Simulink.

Clearly, the Simulink implementation model above is a simplification. In order to really get this to work in Simulink and have the simulation be configurable for different run time conditions, it is necessary to pass the following additional information to the EKF block:

- Simulation time step size Δ
- Initial target state estimate \hat{x}_{int}
- Initial state estimate covariance matrix $P(0|0)$
- Value of q^2
- Value of Bearing Covariance σ_β^2
- Receiver turn rate

Ensuring that the EKF block is being called the proper number of times is instrumental to its proper operation. In order to get the rest of the model to run properly, an advanced solver like “ode4 (Runge-Kutta)” is necessary. This choice of solver, or any other advanced solver, calls the embedded Matlab function block multiple times per time step. The recursive filter will not work properly if it is being called more than once per time step. To solve this problem it is necessary to ensure that the embedded Matlab functions that contain the filter blocks are only being called once per time step. This can be done by selecting “block parameters” to set the sample time to the time step of the simulation.

In order to account for the fact that a TLA in most cases is not able to track through own-ship turns, the measurement noise covariance matrix R is adjusted. When there is no turn, R is set to the measurement noise covariance values that match expected measurement noise received. When the receiver conducts a turn, R is adjusted to be very large. Adjusting R to a large value during a turn makes measurements taken during the turn inconsequential to the filter. It was deemed unnecessary to equip the EKF-specific simulations with TLAs that turned after a towing vehicle or required time to stabilize after a turn. The focus of the EKF-specific simulations was to see how well the EKFs can track a target.

A copy of the code used internally to the BO-EKF block is provided in the Appendix D.

c. Filter Performance

To get a general idea as to how well the BO-EKF performs for state estimation, two specific cases are presented. Both simulation cases have the geometrical encounter described in Table 3. The simulation parameters used in the Case I simulation are contained in Table 4.

The BO-EKF, initialized with the true target state, only really works well when a very large bearing rate is maintained as can be seen by Figure 21. This is due to the target state not being fully observable unless the receiver is out-maneuvering the target [10]. When the receiver is not out-maneuvering the target, the state estimate

Table 3. Simulation parameters used for Case I and Case II simulation runs.

| Item | Value | Units |
|---------------------------|-------|-------|
| Initial Positoin (X-Targ) | 0 | km |
| Initial Positoin (Y-Targ) | 0 | km |
| Target Velocity (X) | 9 | Knot |
| Target Velocity (Y) | 9 | Knot |
| Initial Position (X-Rec) | 0 | km |
| Initial Position (Y-Rec) | 2 | km |
| Reciever Velocity (X) | 8 | Knot |
| Reciever Velocity (Y) | -8 | Knot |
| Turn (Tgt +7.5 min) | R90 | deg |
| Turn (Rec +5 min) | L120 | deg |
| Turn (Rec +9 min) | R90 | deg |
| Simulation Stop Time | 15 | Min |

Table 4. Simulation parameters used for Case I BO-EKF simulation run.

| Item | Value | units |
|-----------------------------------|------------|-------------|
| q^2 | $4\pi/180$ | m^2/sec^2 |
| $\sigma^2(\beta)$ Noise | 1.00E+00 | deg^2 |
| R Matrix $\sigma^2(\beta)$ | 1.00E+00 | deg^2 |
| R Matrix (Turn) $\sigma^2(\beta)$ | 5.73E+07 | deg^2 |
| Tgt Turn Rate | 1.00E+00 | deg/sec |
| Time Step | 1.00E+00 | sec |
| Pint σ^2 (position) | 4.00E+06 | m^2 |
| Pint σ^2 (velocity) | 1.00E+00 | m^2/sec^2 |
| Xint | Xtrue | -- |

diverges. Also, in this case the measurement is very non-linear, which also contributes to the divergence over time. For the BO-EKF, the case where the receiver is maneuvering frequently and driving large bearing rates is probably a best case scenario.

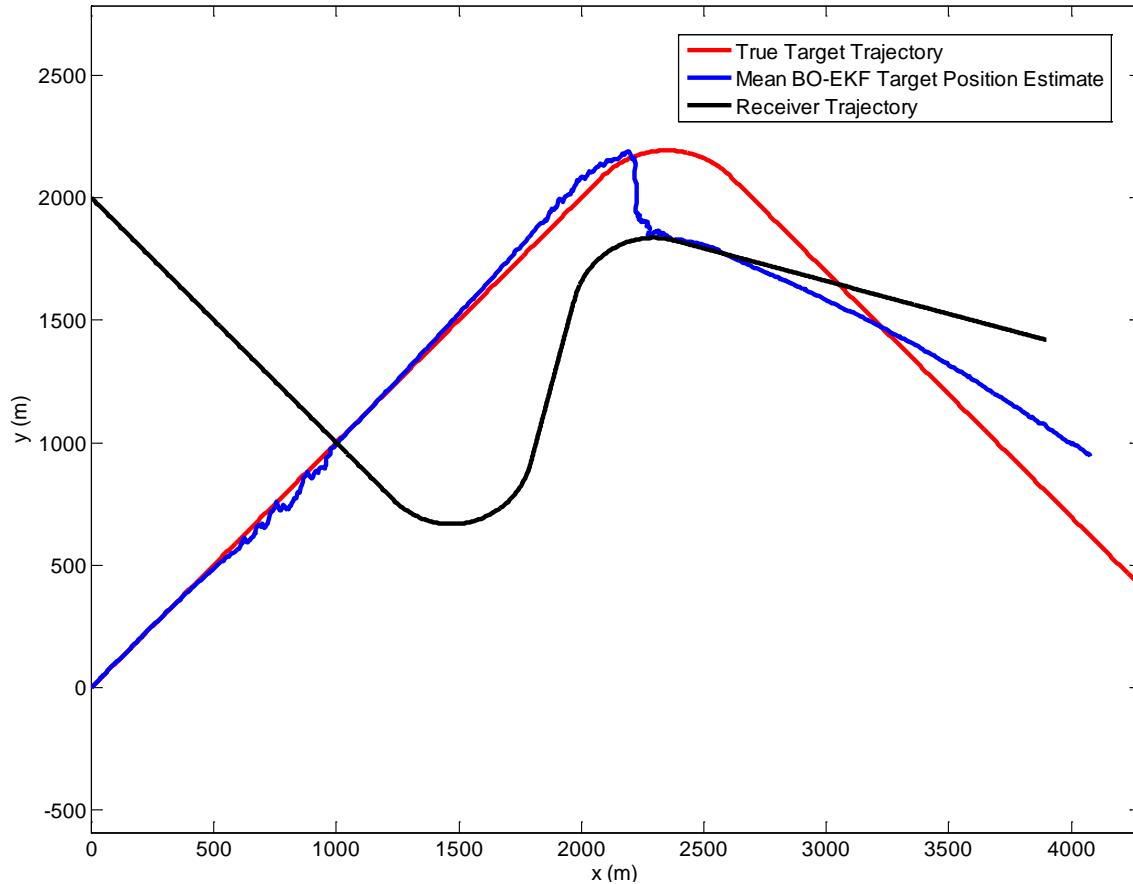


Figure 21. Case I BO-EKF simulation trajectories and position estimate.

Given the ideal conditions for the BO-EKF, there are still large variations in the state estimate, with the range error topping 70% of the total range, as can be seen in Figure 22. It can also be seen that the range error is getting larger in general with time. The range error was also observed over longer simulation times and kept diverging. The only way to prevent further divergence was to conduct large maneuvers to out-maneuver the target platform, and even then, in most cases, it still diverged over time.

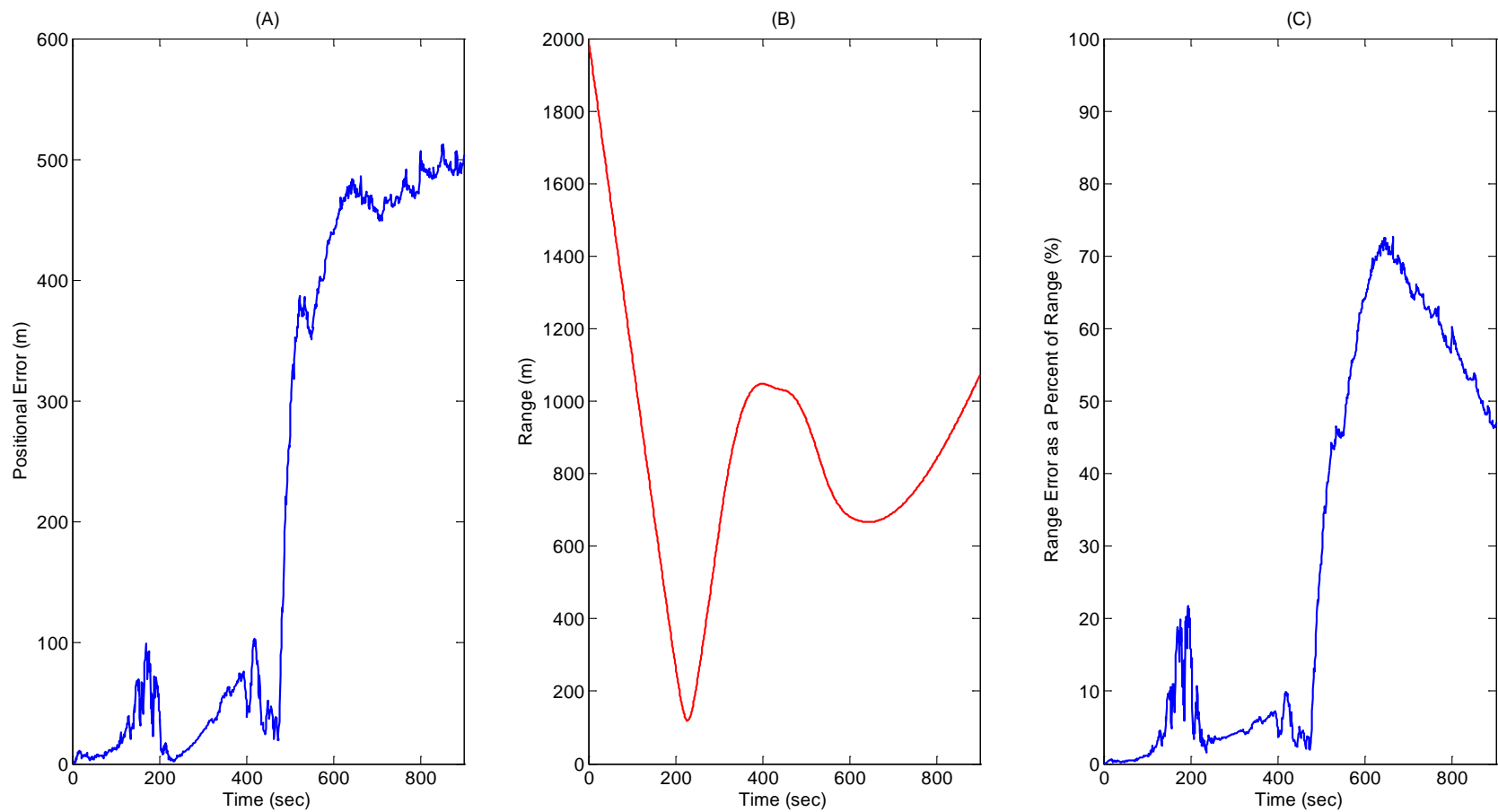


Figure 22. Case I BO-EKF simulation results with target range error over time depicted in (A), total range to target over time depicted in (B), and range error as a percentage of total target range depicted in (C).

The BO-EKF estimated bearings behaved very well for the entire simulation, resulting in less than one degree of bearing error. It is apparent that the bearing estimated for the BO-EKF is smoothing the noisy measured bearing as it should. In line with expectations, the beginning of the simulation had the highest bearing rate and corresponded to the portion of the simulation where the positional errors were minimized.

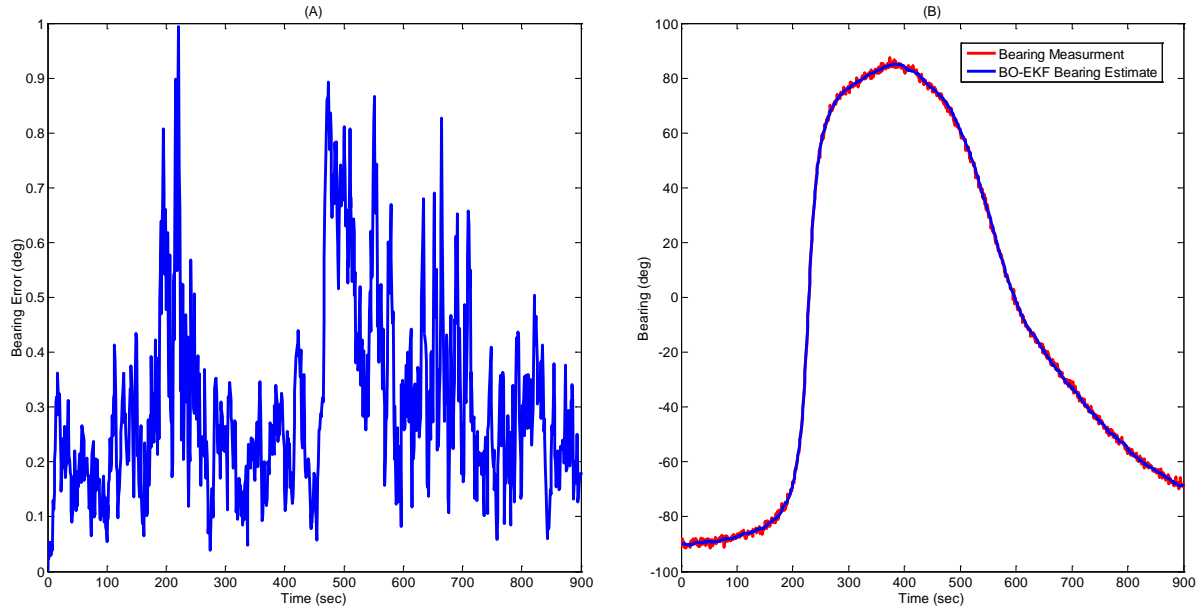


Figure 23. BO-EKF simulation Case I bearing error (A) and measured and estimated bearings over time (B).

In order to show that the poor performance of this state estimation technique is not just a function of some tough geometry, it is beneficial to look at a case where the target and receiver are not maneuvering as much. In Case II, the target and receiver both move in straight line motion in the positive X-direction. The BO-EKF is started with an initial state estimate of truth. Remember that the BO-EKF is expecting that the normal motion exhibited by the target is straight line motion. Intuitively, this would seem like a very easy case where the filter would perform well. Parameters used from case one are the same as used in case two with the exception of no turning.

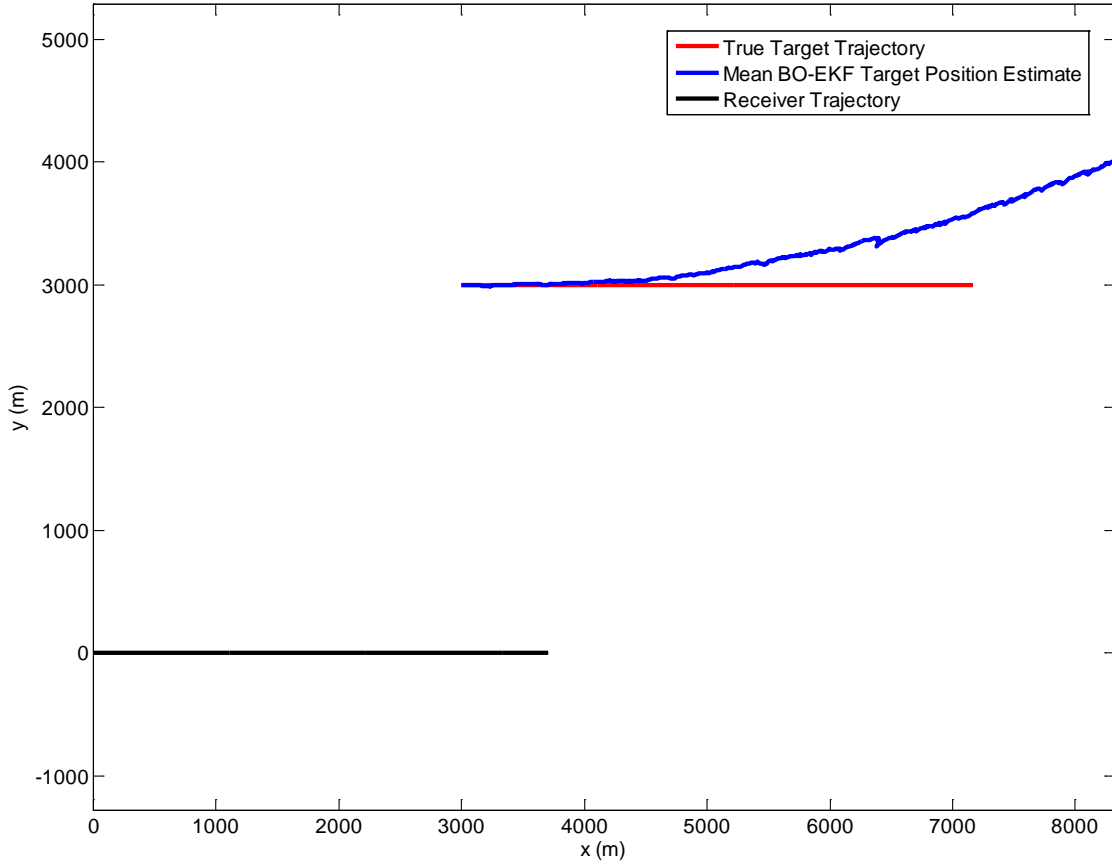


Figure 24. Case II BO-EKF receiver and target trajectories with estimated target position.

Case II, where the motion of the target matches the underlying EKF expectations, is shown in Figure 24. The true state of the target is the initial guess for the target state estimation. Divergence of the state estimate from reality still occurs under these seemingly ideal conditions. This case makes it clear that, because portions of the target state are unobservable, this filter is not the best choice for state estimation.

There is a separate problem that has not been illustrated by these two cases; there is a singularity point in the BO-EKF as written. This occurs when the target goes from a bearing of 180 to -180 degrees. This geometry is fairly easy to avoid in practical applications, but should be protected against. The start of the divergence that occurs when this singularity is traversed is shown in Figure 25.

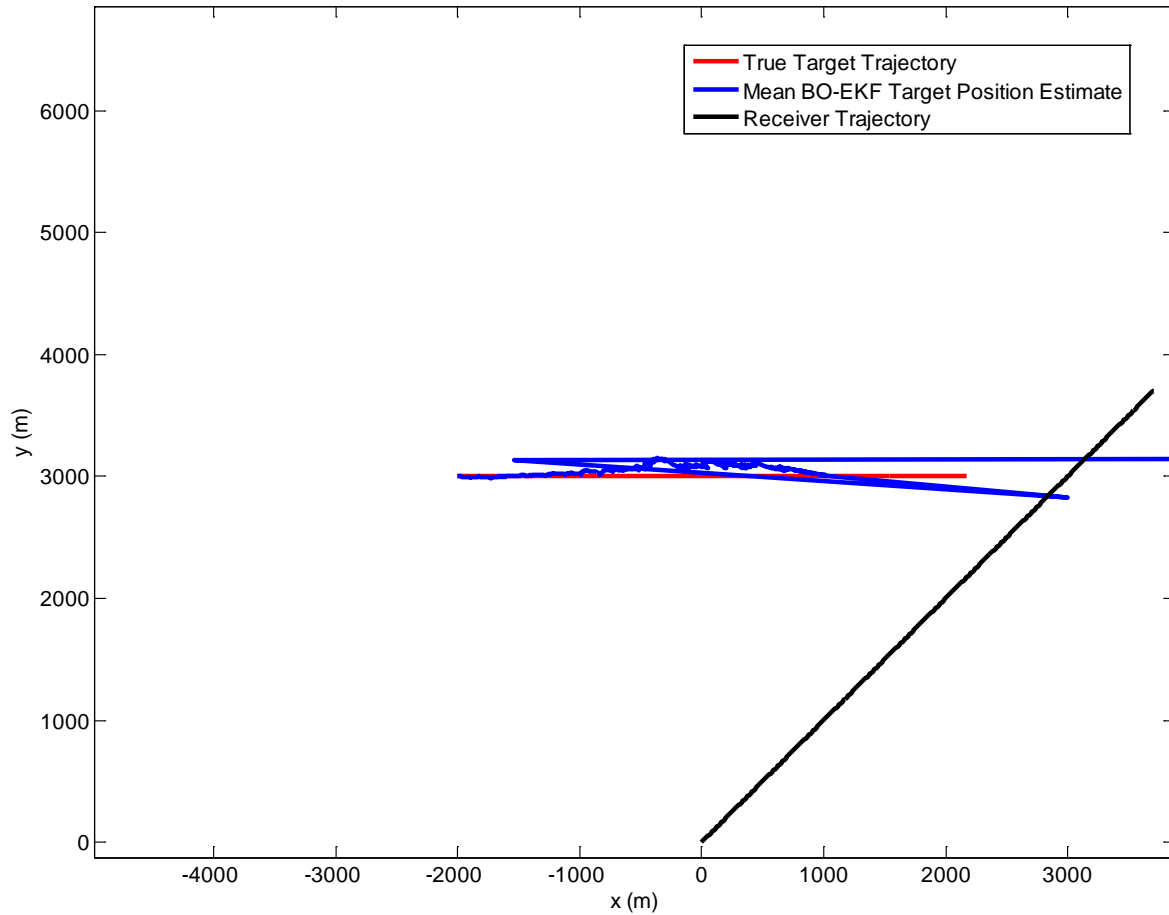


Figure 25. BO-EKF geometry traversing 180 to -180 degree singularity.

The divergence that occurs as a result of traversing the singularity point is very dramatic. The range errors quickly top 2000% of the range to the target as can be seen by Figure 26. The geometry plot was expanded to show the initial divergence.

In order to fix the issue of bearing singularities coming up in specific filter designs, a transition to quaternions is necessary. While using quaternions helps with the singularity, the fix for state estimate divergence with truth needs to be solved by adding information to the filter, making the state more observable. This can be done by adding a component to the filter that gives insight to the range component.

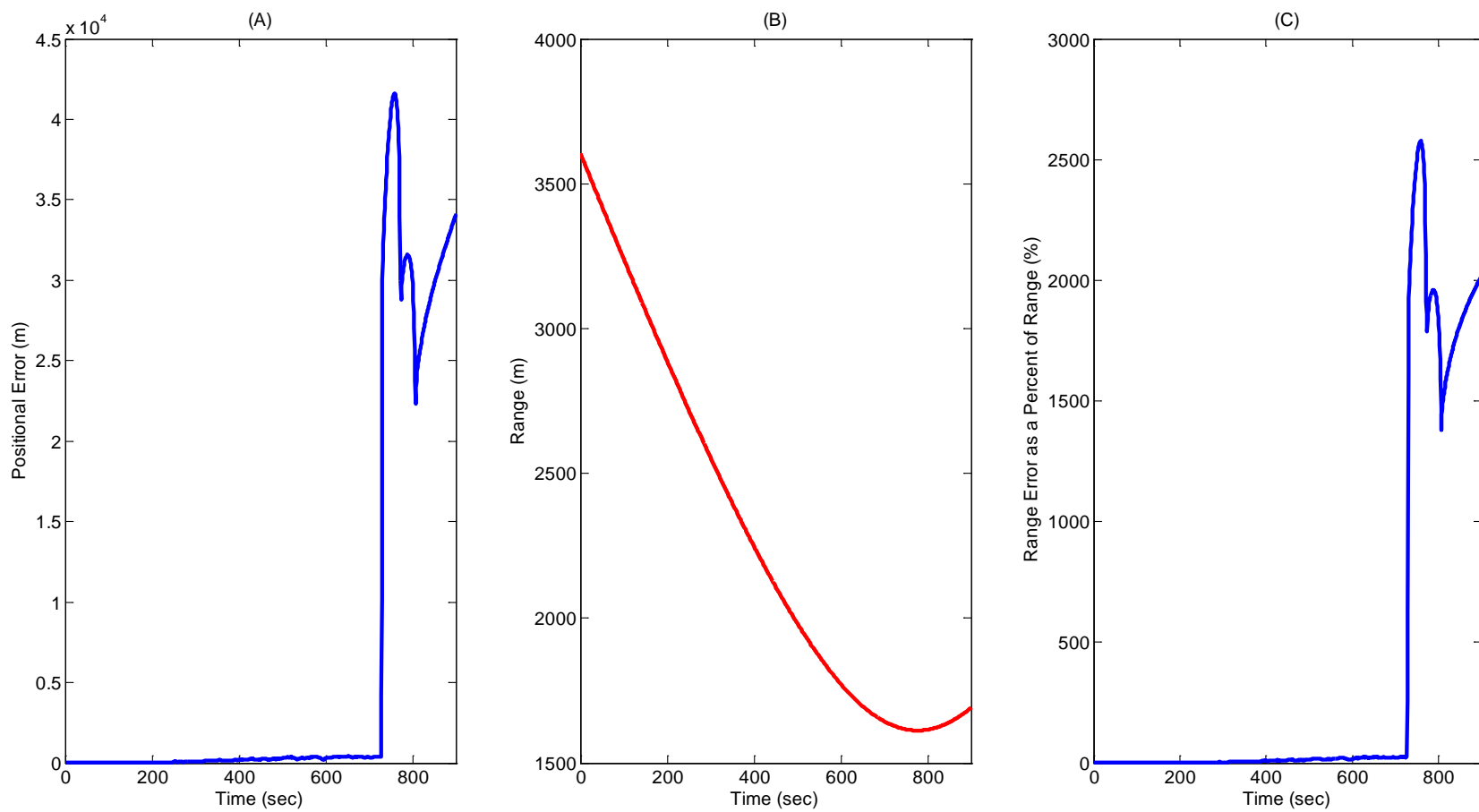


Figure 26. BO-EKF simulation through a singularity point with estimated range error is shown in (A), range to target is shown in (B) and range error as a percentage of total range is shown in (C).

3. Doppler-bearing Tracking Using DB-EKF

a. Overview

Success has been shown using Doppler and Bearing measurements with EKFs of various forms. Generally, measurement terms are linearized resulting in the non-linear terms being grouped into the plant and measurement noise terms. This introduces a bias error that is overcome by using either a Modified Polar coordinate system, Instrumental Variables or a Pseudo Linear Estimator. These methods have been shown to eliminate bias but cannot be used in real time [23]. For a submarine tracking application it is desired that the filter run in real time. The rectangular coordinate system was chosen because of its relative simplicity to implement compared to the solution proposed in [23].

The DB-EKF uses the same process and information as the BO-EKF while adding Doppler information. Targets emit noise in the water, and this noise is received by the TLA in the form of broadband or narrow band tonals. The received narrow band signals, which are indicative of a submerged contact, can be measured with a large degree of precision. The target emits narrow band noise at a frequency. This emitted frequency cannot usually be easily changed as they arise from various machinery or electrical components in use. Since the narrow band signal frequency can be measured with a large degree of precision, one can deduce the direction of relative motion in the line-of-sight. Given a base frequency, if the received frequency is higher than the base frequency, the target's relative motion is closing in the line-of-sight. If the received frequency is lower than the base frequency, the target is opening in the line-of-sight. It is necessary to guess or estimate a base frequency of the received noise. In order to make a good guess at the base frequency being emitted in the water by a contact, one must take into account any available intelligence on the source and bounding characteristics on target speed.

The state vector for the DB-EKF is given by

$$\vec{x}_t = \begin{bmatrix} x_t \\ v_{xt} \\ y_t \\ v_{yt} \\ f_b \end{bmatrix} = \begin{bmatrix} x - position \\ x - velocity \\ y - position \\ y - velocity \\ frequency - base \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}. \quad (29)$$

Note that the state vector contains the base frequency of the target signal. The receiver state is still known at all times and was given in Equation (16). The process noise covariance matrix is given by

$$Q(k) = \begin{bmatrix} \frac{q^2 \Delta^3}{3} & \frac{q^2 \Delta^2}{2} & 0 & 0 & 0 \\ \frac{q^2 \Delta^2}{2} & q^2 \Delta & 0 & 0 & 0 \\ 0 & 0 & \frac{q^2 \Delta^3}{3} & \frac{q^2 \Delta^2}{2} & 0 \\ 0 & 0 & \frac{q^2 \Delta^2}{2} & q^2 \Delta & 0 \\ 0 & 0 & 0 & 0 & q_f^2 \Delta \end{bmatrix} = q^2 \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (30)$$

Here q_f^2 is a coefficient representing the unknown shifting in the base frequency. The base frequency is assumed to not change significantly over time, thus q_f is given a value of zero. As stated before, the base frequency should not change over time unless there is an active effort being made by the target to alter the target frequency.

The measurement in this case is the bearing and received frequency, and the measurement matrix is given by

$$z(k) = h(\vec{x}) + \omega(k) = \begin{bmatrix} \beta(k) \\ f_r \end{bmatrix} + \omega(k) = \begin{bmatrix} \arctan\left(\frac{y_t(k) - y_r(k)}{x_t(k) - x_r(k)}\right) - Hdg \\ f_b \left(1 + \frac{V_{los}}{ss}\right) \end{bmatrix} + \omega(k), \quad (31)$$

where ss is the speed of sound in water, and V_{los} is the velocity in the line-of-sight and is given by

$$V_{los} = (v_{rx} - v_{tx}) \cos(\beta) + (v_{ry} - v_{ty}) \sin(\beta) = \frac{(v_{rx} - v_{tx})(x_t - x_r) + (v_{ry} - v_{ty})(y_t - y_r)}{\sqrt{(x_t - x_r)^2 + (y_t - y_r)^2}}. \quad (32)$$

Equation (32) can be further simplified by defining a few more terms. The relative X and Y velocities are given by

$$\Delta V_x = (v_{rx} - v_{tx}), \quad (33)$$

and

$$\Delta V_y = (v_{ry} - v_{ty}). \quad (34)$$

The range from the receiver platform to the target platform is given by

$$Rng = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2}. \quad (35)$$

Substituting Equations (33), (34) and (35) into Equation (32), we get

$$V_{los} = \Delta V_x \cos(\beta) + \Delta V_y \sin(\beta) = \frac{(\Delta V_x)(x_t - x_r) + (\Delta V_y)(y_t - y_r)}{Rng}. \quad (36)$$

It is important to note that a closing velocity in the line-of-sight has been chosen to be described by a positive value and an opening velocity in the line-of-sight as a negative value.

The measurement estimate for time k given $k-1$ measurements is given by

$$\hat{z}(k | k-1) = h(k, \hat{x}_t(k | k-1), \bar{x}_r) = \begin{bmatrix} \hat{\beta}(k | k-1) \\ \hat{f}_R(k | k-1) \end{bmatrix} = \begin{bmatrix} \arctan \left(\frac{\hat{y}_t(k | k-1) - y_r(k)}{\hat{x}_t(k | k-1) - x_r(k)} \right) - Hdg \\ \hat{f}_B(k | k-1) \left(1 + \frac{\hat{V}_{los} \left(k, \hat{x}(k | k-1), \bar{x}_r \right)}{ss} \right) \end{bmatrix}. \quad (37)$$

Therefore, the gradient of $h(\bar{x}_t, \bar{x}_r, k)$ with respect to \bar{x}_t is given by

$$\frac{\partial h'(\vec{x}_t, \vec{x}_r, k)}{\partial \vec{x}_t} = H'(\vec{x}_t, \vec{x}_r, k) = \begin{bmatrix} \frac{-\Delta Y}{Rng^2} & \frac{f_B(\Delta Y) \{ (\Delta V_x)(\Delta Y) - (\Delta V_y)(\Delta X) \}}{(ss)(Rng)^3} \\ 0 & \frac{f_B(\Delta X)}{(ss)(Rng)} \\ \frac{\Delta X}{Rng^2} & \frac{f_B(\Delta X) \{ (\Delta V_y)(\Delta X) - (\Delta V_x)(\Delta Y) \}}{(ss)(Rng)^3} \\ 0 & \frac{f_B(\Delta Y)}{(ss)(Rng)} \\ 0 & 1 + \frac{V_{los}}{ss} \end{bmatrix} \quad (38)$$

where ΔX and ΔY are given by

$$\Delta X = (x_t - x_r) \quad (39)$$

and

$$\Delta Y = (y_t - y_r). \quad (40)$$

The filter progresses through time in the same manner as described for the BO-EKF shown in Equations (24) - (28).

b. DB-EKF Model

The recursive DB-EKF algorithm is coded in an embedded Matlab function block shown in Figure 27.

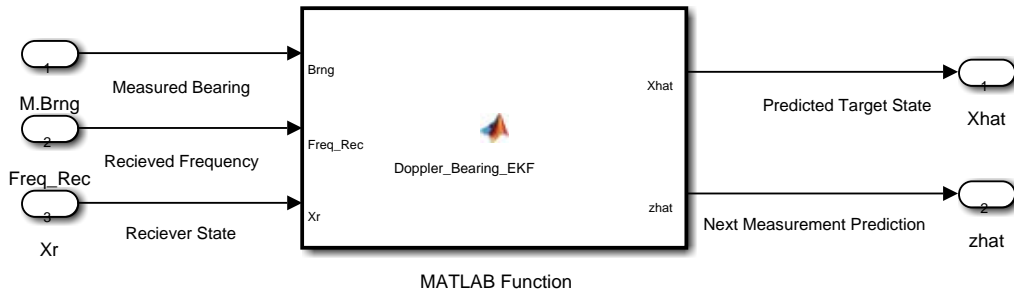


Figure 27. A depiction of a generic DB-EKF Embedded Matlab Function block.

Clearly, this depiction of a generic DB-EKF Embedded Matlab Function block represents a simplification of what is really necessary to implement the DB-EKF. To implement the DB-EKF, it is necessary to pass the following additional information to the filter block:

- Simulation time step size Δ
- Initial target state estimate \hat{x}_{int}
- Initial state estimate covariance matrix $P(0|0)$
- Value of q^2 and q_f^2
- Value of bearing covariance σ_β^2
- Value of received frequency covariance $\sigma_{f_R}^2$
- Speed of sound in receiver medium ss
- Receiver turn rate

A copy of the internal code used for the DB-EKF and the actual Simulink layout used is provided in Appendix D.

c. Filter Performance

To determine how the DB-EKF performs compared to the BO-EKF, cases one and two are run with both types of filters operating. The cases are simulated one hundred times, and the mean values for trajectories and errors are shown. The parameters for case one are shown in Table 5. The starting geometrical setup and trajectories for the receiver and target platforms are described in Table 6.

The result of the simulation showing the mean estimated target positions for the DB-EKF and BO-EKF compared to the actual target trajectory are captured in Figure 28.

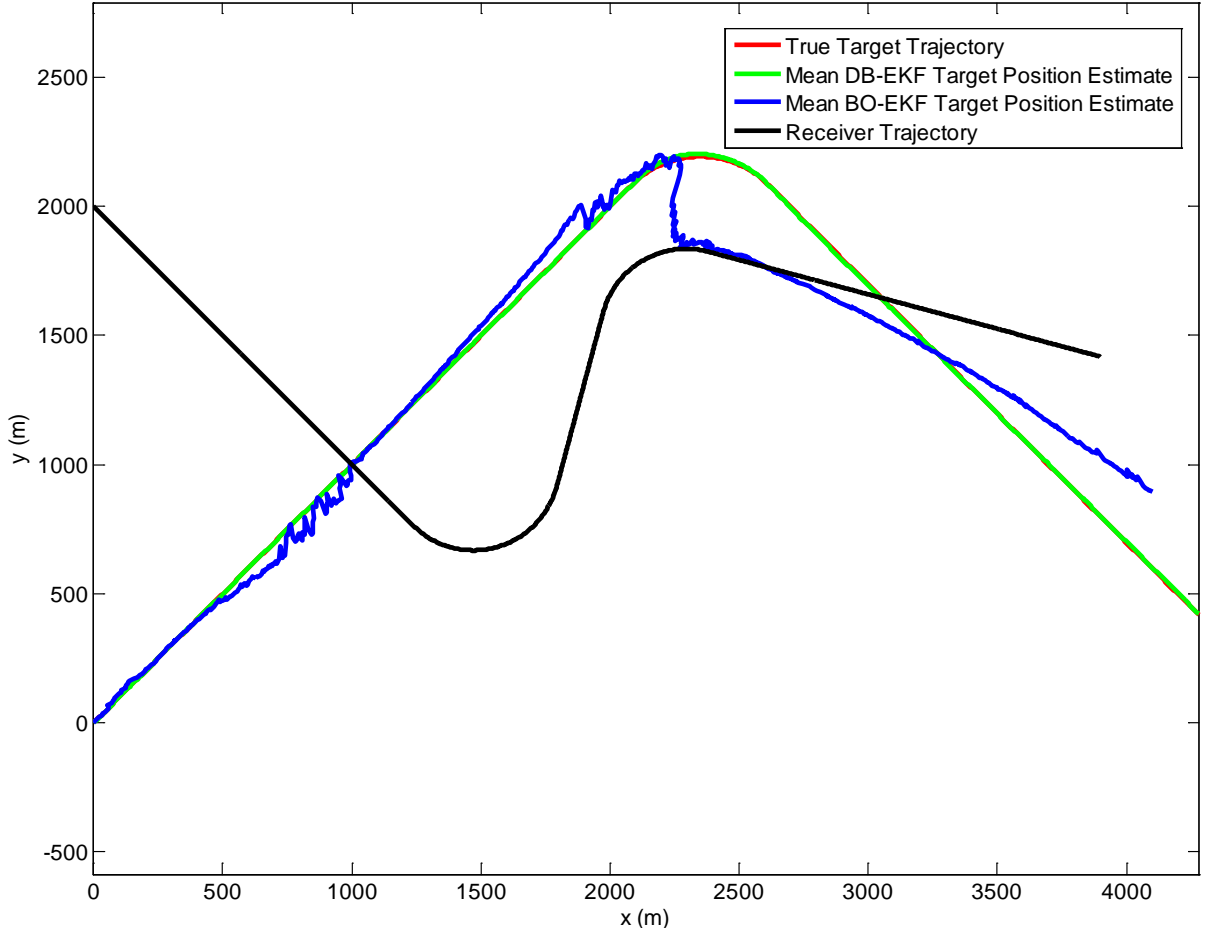


Figure 28. Case I DB-EKF and BO-EKF mean estimated target positions with actual target and receiver trajectories.

The DB-EKF clearly outperforms the BO-EKF, even when the receiver is maneuvering more than the target. In this case both filters are initiated with the true target state. The range errors are shown in Figure 29.

For Case I, where the mean BO-EKF range error ratio peaks at over 70% the target range, the mean DB-EKF range error ratio does not exceed 10%. Because the receiver outmaneuvers the target in Case I, it should yield good BO-EKF performance. Even with an ideal case for BO-EKF performance, the DB-EKF outperforms the BO-EKF.

Table 5. Case I simulation parameters for DB-EKF and BO-EKF comparison.

| Item | Value | units |
|-----------------------------------|------------|-------------|
| q^2 | $4\pi/180$ | m^2/sec^2 |
| q_{fB}^2 | 0.00E+00 | $1/sec^3$ |
| $\sigma^2(\beta)$ Noise | 1.00E+00 | deg^2 |
| $\sigma^2(f_R)$ Noise | 4.50E-07 | hz^2 |
| R Matrix $\sigma^2(\beta)$ | 1.00E+00 | deg^2 |
| R Matrix $\sigma^2(f_R)$ | 4.50E-07 | hz^2 |
| R Matrix (Turn) $\sigma^2(\beta)$ | 5.73E+07 | deg^2 |
| R Matrix (Turn) $\sigma^2(f_R)$ | 1.00E+06 | hz^2 |
| Tgt Turn Rate | 1.00E+00 | deg/sec |
| Time Step | 1.00E+00 | sec |
| Pint $\sigma^2(\text{position})$ | 4.00E+06 | m^2 |
| Pint $\sigma^2(\text{velocity})$ | 1.00E+00 | m^2/sec^2 |
| Xint | Xtrue | -- |

Table 6. Case I trajectory description for DB-EKF and BO-EKF comparison.

| Item | Value | Units |
|---------------------------|-------|-------|
| Initial Positoin (X-Targ) | 0 | km |
| Initial Positoin (Y-Targ) | 0 | km |
| Target Velocity (X) | 9 | Knot |
| Target Velocity (Y) | 9 | Knot |
| Initial Position (X-Rec) | 0 | km |
| Initial Position (Y-Rec) | 2 | km |
| Reciever Velocity (X) | 8 | Knot |
| Reciever Velocity (Y) | -8 | Knot |
| Turn (Tgt +7.5 min) | R90 | deg |
| Turn (Rec +5 min) | L120 | deg |
| Turn (Rec +9 min) | R90 | deg |
| Simulation Stop Time | 15 | Min |

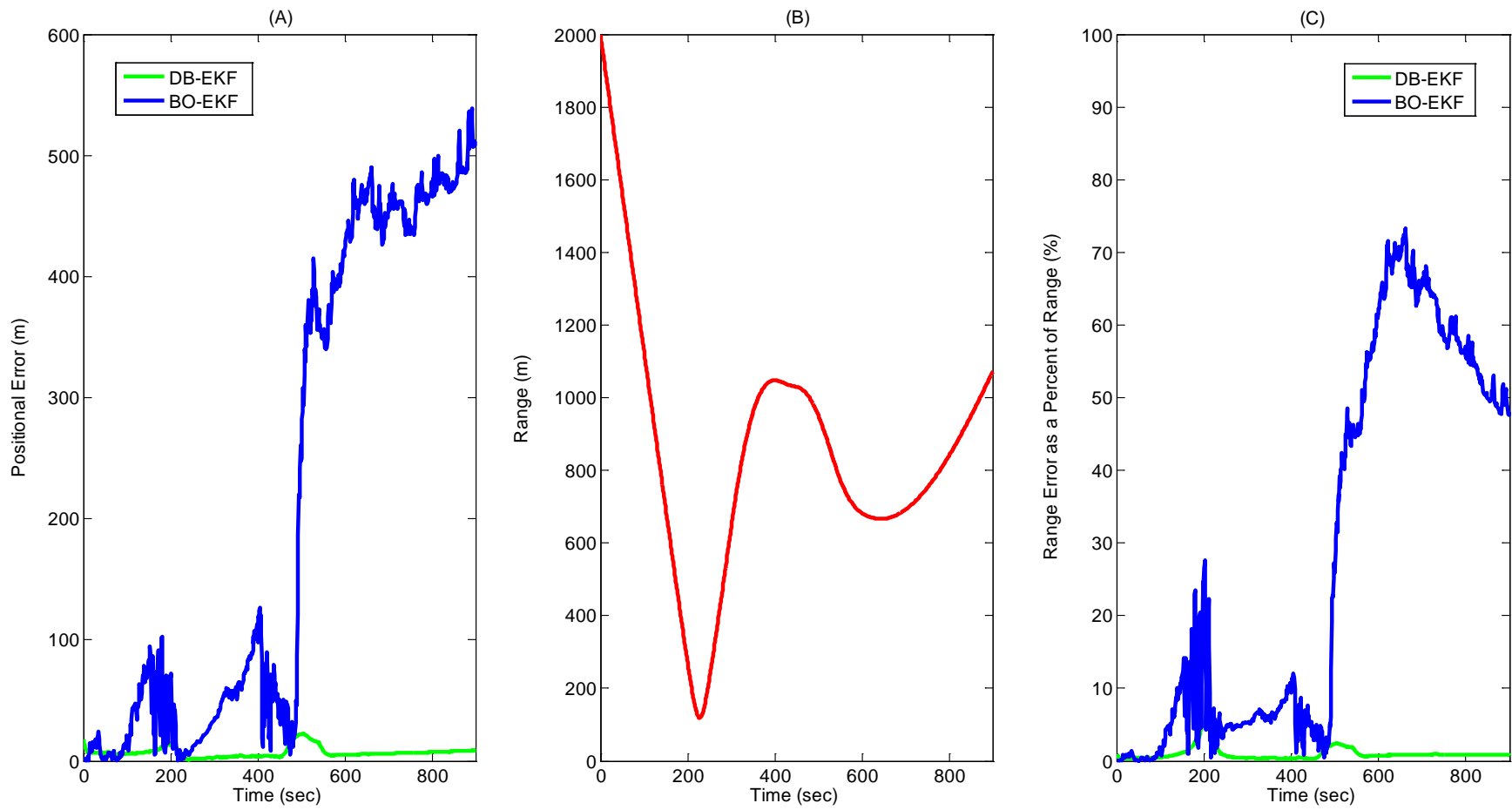


Figure 29. Case II simulation for DB-EKF and BO-EKF showing mean range errors (A), range to target platform (B), and range error as a percentage of total range (C).

The measured bearings, mean estimated bearings, and mean bearing errors are shown in Figure 30. It can be seen that the mean bearing error is better in the DB-EKF with the exception of when the target is maneuvering. The mean bearing errors for the DB-EKF case are much more stable in general. Looking at the plot of the measured bearings and mean estimated bearings for the DB-EKF and BO-EKF, it is apparent that both filters are dealing with the noisy bearings in an acceptable manner.

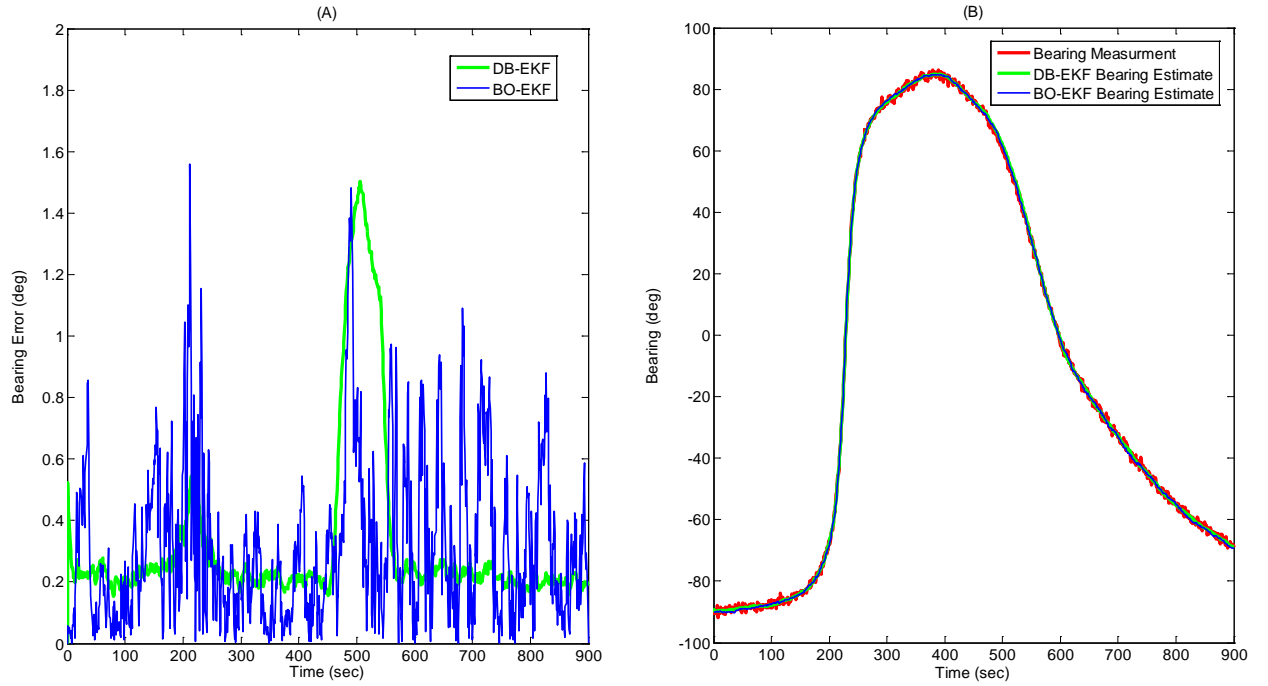


Figure 30. Case II DB-EKF and BO-EKF bearing error over time (A), and measured and estimated bearings over time (B).

Clearly, the DB-EKF outperforms the BO-EKF when the receiver is maneuvering. To determine if it performs as well or better than the BO-EKF when the receiver platform is not maneuvering, Case II is detailed. In Case II, the filter parameters stay the same, and the target and receiver platform move in straight line motion for fifteen minutes. The mean trajectories for one hundred simulation runs with both filters are shown in Figure 31. It is immediately obvious that the DB-EKF performs better when there is little relative motion. Utilizing the additional Doppler information, we see that it is no longer necessary for the receiver platform to out-maneuver the target platform to enable

convergence of the filter. The DB-EKF still has a singularity that will cause problems if the target traverses 180 degrees relative to the receiver. As previously stated, the use of quaternions would alleviate this problem.

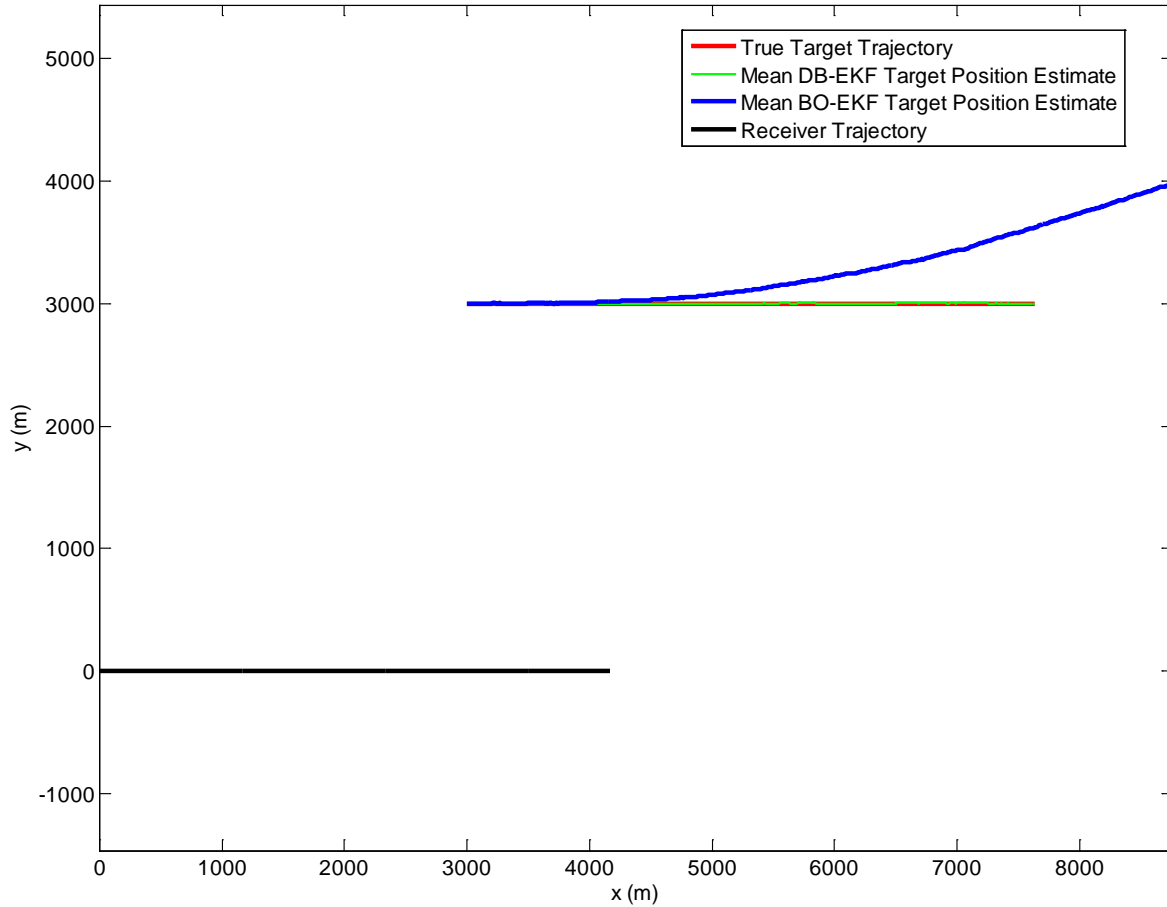


Figure 31. Case II target and receiver trajectories with mean estimates for target position from DB-EKF and BO-EKF.

Under normal circumstances, filters cannot be initiated with the true state of the target. Therefore, it is necessary to see how these filters perform when there is a bad initial estimate on position and base frequency. Case IA is a revisiting of Case I with parameters shown in Table 6. There is an initial positional error of 1000 meters down the bearing of the target. This is a very large positional error, in this case representing an additional 50% of the total range to the target platform. The simulation results of Case IA are presented in Figure 32.

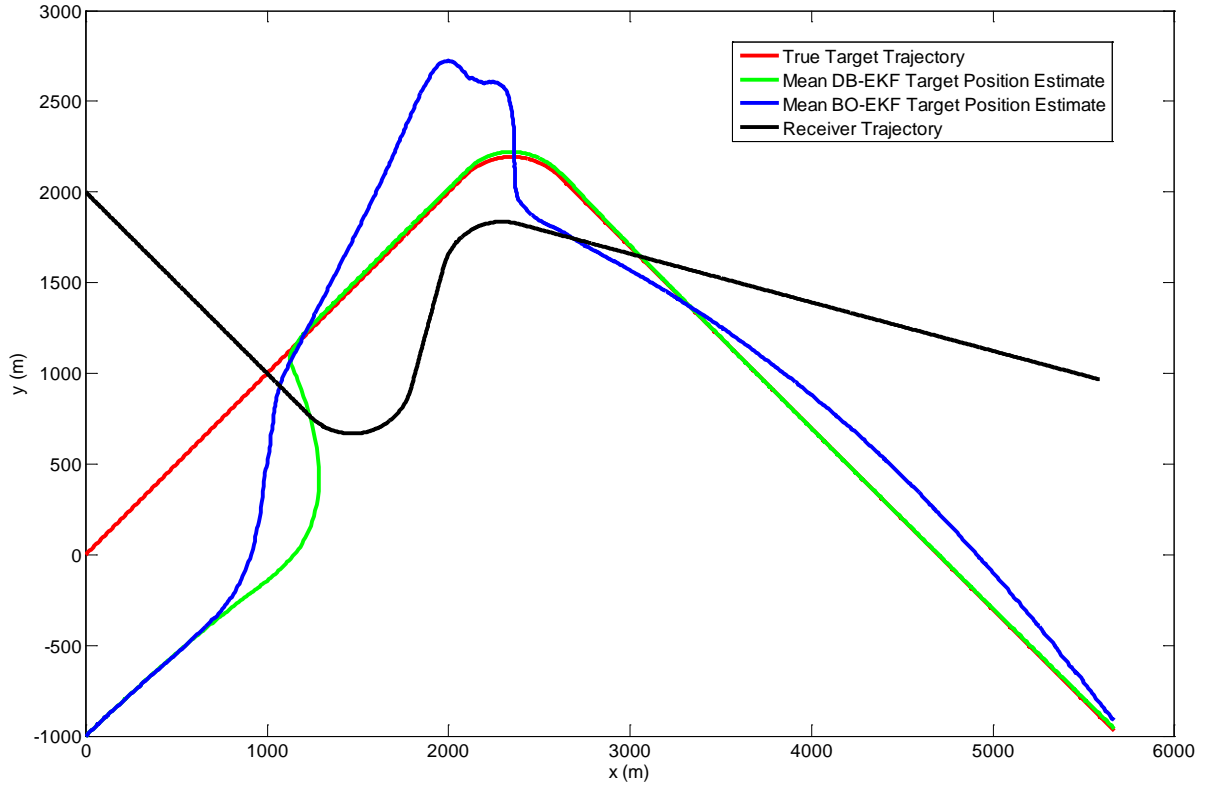


Figure 32. Case IA target and receiver trajectories with mean estimates for target position from DB-EKF and BO-EKF.

The initial position estimates are along the same bearing as the target but at 150% the real range. The DB-EKF settles on the correct trajectory and tracks well. Unfortunately, this is not always the behavior that is observed. For instance, if in this case the initial range estimate is the true range reduced by 50%, a very different behavior is observed. This has been simulated as Case IB and plotted in Figure 33.

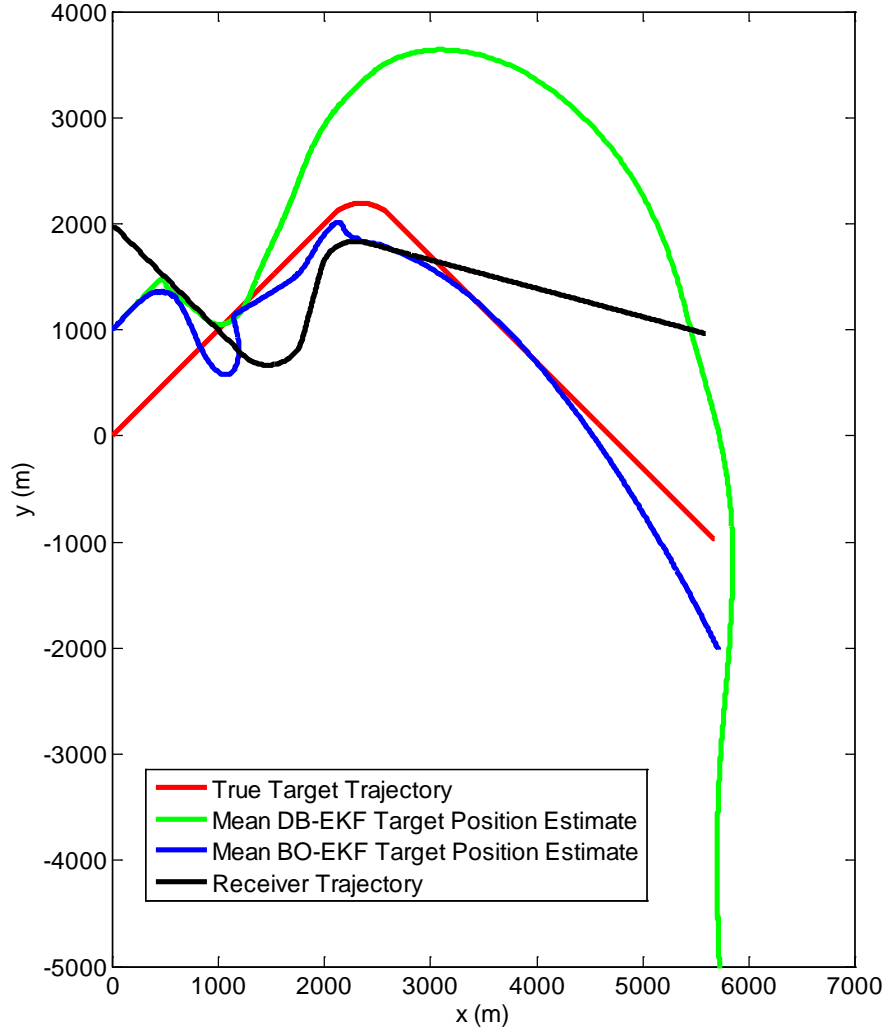


Figure 33. Case IB target and receiver trajectories and mean estimated target positions for DB-EKF and BO-EKF.

With the initial estimated range equal to half of the real range, the DB-EKF does not converge to the target trajectory. This seems to be mainly due to the way that the base frequency is treated when the bearing of the target shifts dramatically. The large estimated base frequency shift is shown in Figure 34. The state covariance matrix P should be monitored to ensure that the state estimate is not devolving as in this case. The rate of change and magnitude of change in the base frequency estimate could also be monitored to ensure that this behavior does not occur.

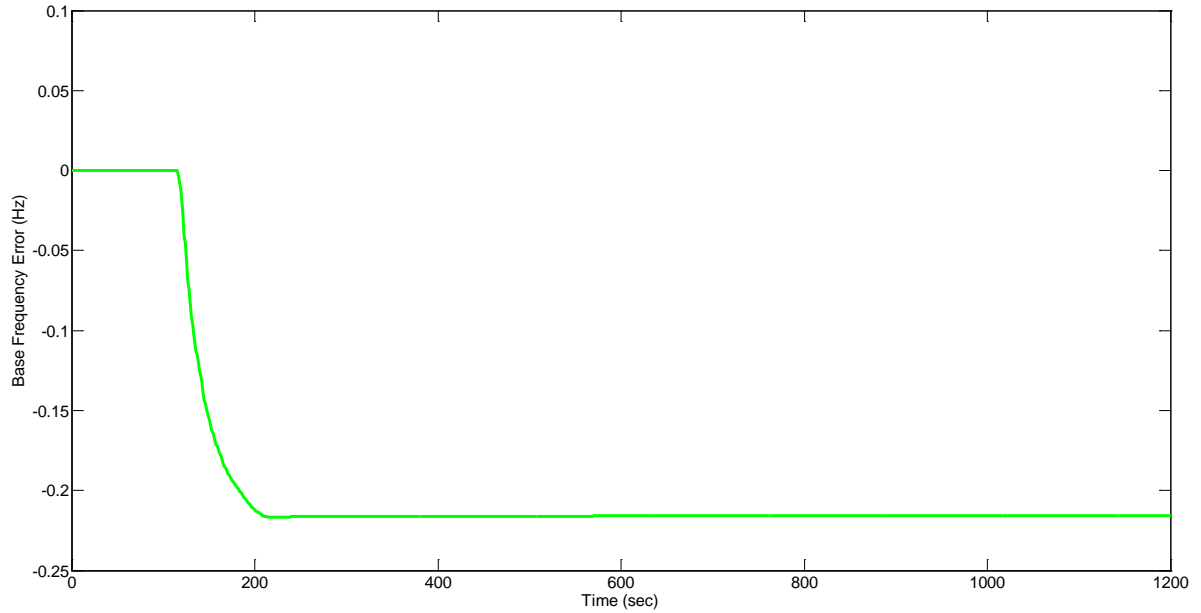


Figure 34. Case IB DB-EKF base frequency error over time.

Unfortunately, there is more bad news when it comes to using a DB-EKF for target tracking. There is an issue with biasing. This is discussed in [23] with solutions presented. To see the effect of biasing on the DB-EKF, one only needs to alter the geometry of Case IA a bit. Simulation of a case where the range to the target is 8 km at the onset, with an initial range estimate short of the actual range, is shown in Figure 35. Even though the DB-EKF follows the motion of the target, there is still a bias that is not reduced significantly over time.

This can be overcome by the choice of receiver platform maneuver. By driving a large amount of bearing rate, one can shrink the bias over time. This is most easily accomplished by crossing the target's line-of-sight. This will cause a large bearing change even when the target is at a large range. The result of such a maneuver chosen to drive a large bearing rate is shown in Figure 36.

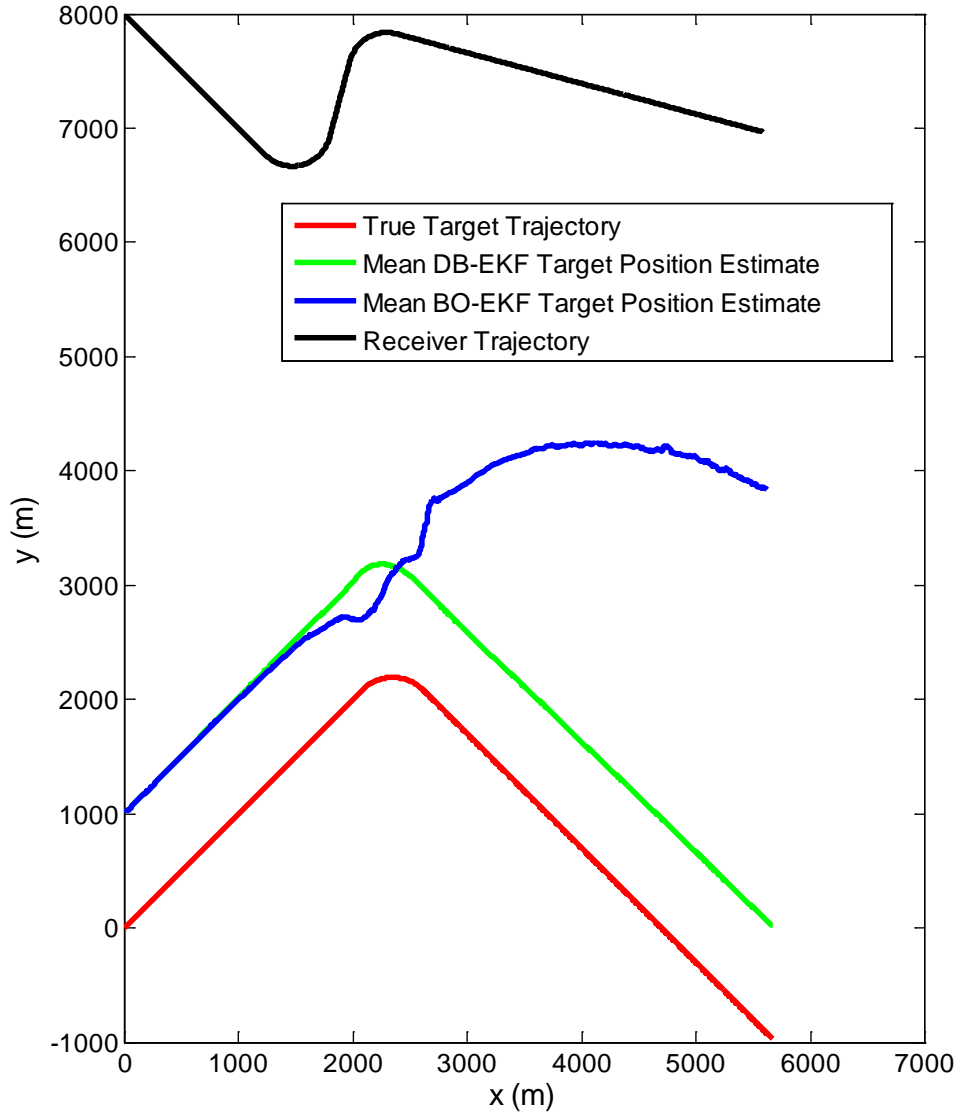


Figure 35. True target and receiver trajectories with mean estimated positions using DB-EKF and BO-EKF with biasing.

Driving a large bearing rate helps both the BO-EKF and DB-EKF converge onto the target trajectory and is essentially what [23] referred to as out-maneuvering the target platform.

It is important to remember that even though the DB-EKF has been shown to have some weaknesses, it performs in a superior manner when compared to BO-EKF and can be used to track targets in a multitude of situations.

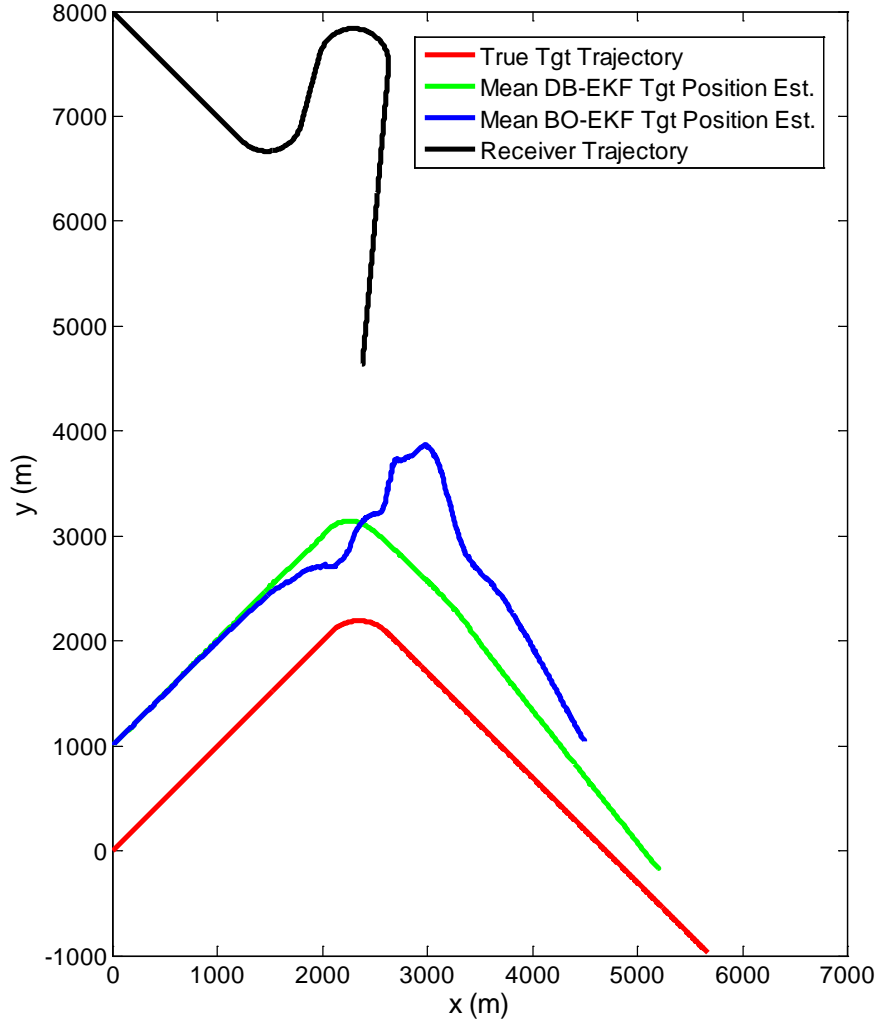


Figure 36. True target and receiver trajectories with DB-EKF and BO-EKF mean estimated positions, where biasing is overcome by increasing bearing rate.

4. Omission Choice for SNR Information for EKF

Tracking performance increased significantly when the Doppler information was included with the bearing information. TLAs, in general, give the user one more piece of information that could potentially be used to improve tracking. This extra piece of information is the SNR of the received signal. On the surface it would seem that signal strength would always have an inverse relationship with target range, but in real world environments this is not the case. SNR can increase while range is not decreasing under

a multitude of conditions that are regularly seen at sea. Examples where SNR could increase as range increases include, but are not limited by:

- Entering convergence zone
- Target noise level increasing
- Transition to a more reflective bottom type
- Decrease in ambient noise
- Decrease in receiver platform noise
- Target bearing unmerging with interfering contact bearing
- Decrease in scattering as biologics (fish schools) are no longer between target and receiver
- Boundary layer transition

With many cases where SNR does not follow an inverse relationship with range, the decision has been made to not include SNR information into the filter. To simulate how this information would affect actual filter performance would require modeling all the various effects of the complicated ocean environment, which is not the focus of this thesis. While SNR information is usually not incorporated into tracking filters, it is used by experienced operators in tracking scenarios to glean more information while having access to indications of the various reasons for the change.

D. CHAPTER SUMMARY

In this chapter the BO-EKF and DB-EKFs were presented and their feasibility for use for submarine tracking analyzed. The reasoning for including the Doppler information but omitting the SNR information was discussed. Limitations and issues with both the BO-EKF and DB-EKF were identified, and solutions to mitigate problems were identified. In the next chapter, how the receiver and target platform motion was modeled in Simulink and what kind of environment was modeled to simulate the target and receiver platform interactions are discussed.

IV. MOTION AND ENVIRONMENT MODELING

A. MODELING TARGET AND RECEIVER MOVEMENT

1. Platform Motion

Motion of the target and receiver platform is simulated using a continuous time state space model with constant velocity [24, 22]. The target and receiver platform state equations are

$$\dot{\bar{X}}(t) = \begin{bmatrix} \dot{x} \\ \dot{V}_x \\ \dot{y} \\ \dot{V}_y \end{bmatrix} = A \bar{X}(t), \quad (41)$$

where A represents the system matrix and in the case where velocity is constant, is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\Omega \\ 0 & 0 & 0 & 1 \\ 0 & \Omega & 0 & 0 \end{bmatrix}. \quad (42)$$

Here Ω represents the turn rate with a clockwise turn being positive. This is modeled in Simulink by using the Embedded Matlab Function with the setup shown in Figure 37.

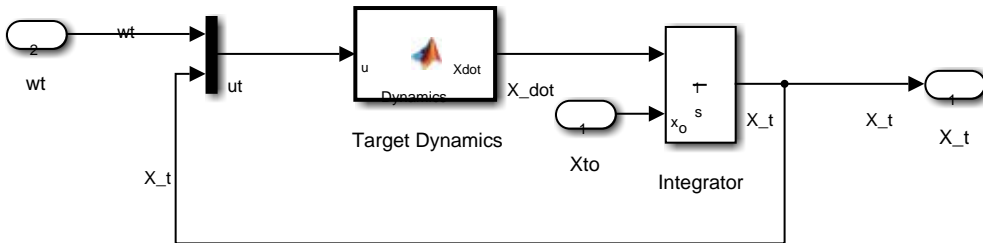


Figure 37. Target and receiver platform motion modeling.

The Dynamics block is a coded version of Equation (41). Because A depends upon the turn rate, it is necessary to pass the Dynamics block the value of the turn rate in addition to the state vector.

2. Sensor Motion

The TLA is towed by the receiver platform, and the output of the TLA depends upon the position and relative geometry between the various hydrophones in the TLA. The individual hydrophone motion is modeled by following exactly, or with an input perturbation, the motion of the receiver platform with a time delay. This time delay is given by

$$\Delta t_{H_n} = \frac{\left(Scope_{TLA} + \frac{AR_{Length}}{2} - nd \right)}{V_{platform}}, \text{ for odd } N, \quad (43)$$

and

$$\Delta t_{H_n} = \frac{\left(Scope_{TLA} + \frac{AR_{Length}}{2} - sign(n) \left(\frac{d}{2} \right) - nd \right)}{V_{platform}}, \text{ for even } N \quad (44)$$

where

$$n = \left[-\frac{N-1}{2} \quad \dots \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad \dots \quad \frac{N-1}{2} \right], \text{ for odd } N \quad (45)$$

and

$$n = \left[-\frac{N}{2} \quad \dots \quad -2 \quad -1 \quad 1 \quad 2 \quad \dots \quad \frac{N}{2} \right], \text{ for even } N. \quad (46)$$

This positional delay is implemented in Simulink by using the Variable Time Delay blocks with the correct time delay input for each hydrophone placed on the position output of the receiver platform.

B. MODELING OCEAN ENVIRONMENT

The ocean environment, along with the lack of received data, is what truly make submarine tracking difficult. The ocean environment has many properties that make submarine tracking complicated and include, but are not limited to, the following [8]:

- Dynamic Sound Speed Profile (SSP)
 - Varies with temperature, pressure, salinity
- Varying bottom types
- Multipath acoustic signal transmission
 - Convergence zones
 - Presence of signal harmonics
- Interfering noise sources (man-made, biologic, and weather related)
- Dynamic scattering environment
- Well defined boundary layers
 - Acoustic signal ducting
 - Shadow zones
- Frequency dependent acoustic signal attenuation
- Signal spreading losses

Because depth is usually not a large portion of range to a target platform, and for model simplicity, only the X and Y dimensions were modeled. This means that the depth dependent properties of the ocean environment were not captured. It was assumed that the SSP was given by a constant uniform speed of sound. The direct path was the only path from the target to the platform that is modeled. Signal attenuation was modeled to be only a function of spherical spreading losses. The Doppler shift of the signal was produced by determining the speed in the line-of-sight and shifting the transmitted signal accordingly. This is accomplished in Simulink by passing the velocity in the line-of-sight through a Voltage Controlled Oscillator Block [25] with the Quiescent Frequency set as the base frequency and the Sensitivity set as the Base Frequency divided by the Speed of Sound. The model used for signal transmission, Doppler shifting and attenuation is attached in Appendix D.

C. CHAPTER SUMMARY

How platform and hydrophone motion is modeled and how the ocean environment was modeled was discussed in this chapter. Conclusions and recommendations for future research to improve U.S. Navy ASW capabilities are contained in the next chapter.

V. CONCLUSIONS AND FUTURE RESEARCH

A. CONCLUSIONS

The generation of dynamic TLA and state estimation models in Simulink were detailed in this thesis. The U.S. Navy could use these models, and future improvements of them, to assist in meeting future ASW goals. These models should form a base platform that can be expanded to assist in improving underwater target tracking.

First, the generation of a dynamic TLA model was detailed which used Dolph-Chebyshev Optimization to form specified beams. This resulted in a model which could be used to simulate small and large arrays alike. While it is desirable to have flexibility in towed array sizes modeled, the penalty in the form of the time required to complete a simulation using this method was excessive for long arrays. This model is successful when used to determine if a specified configuration would work well. The information gained could then be used to build a specific model, which could more quickly simulate an array of a specified size.

Next, target state estimation techniques were detailed. The BO-EKF and DB-EKF were implemented and the benefits and consequences of their usage were explored. A discussion of why these filtering techniques are probably not the best underwater target tracking solution should prove informative and should be used to steer future work to more useful filtering techniques.

B. FUTURE RESEARCH

Dynamic TLA modeling can be used to best match a towed array with a specific towing platform. It can also be used to feed information to state estimation models and eventually be used to close a control loop which could direct the towing platforms movement based upon the received signals.

When trying to bring UUVs, USVs, and surface ships into the ASW arena, TLA models will be vital in determining how to best configure and employ the systems. As

UUVs and USVs become more prevalent, it will be important to derive their control algorithms from complex tracking models.

This work can be specifically improved by speeding up the TLA model in such a manner that it can be efficiently tied to the input of the state estimation models. This would allow motion control algorithms to be developed and tested. A specific recommendation is to use optimal control or potential field algorithms to steer a tracking vessel into a dynamically determined preferred tracking position. With the use of heuristics on ship movement and detection parameters, a means to safely track very quiet submerged targets in a safe manner could be derived. Currently, there seems to be a large amount of risk involved in tracking a quiet, submerged target with a sensor that requires maneuvers for good data while at the same time loses the ability to provide data during, and after, said maneuver. The U.S. submarine force has adopted methods to maintain safety of ship for various other unsafe situations, like look-intervals for periscope operations, and it would seem that similar heuristic work in submarine TLA-only tracking would be warranted.

Improvements to the TLA model could also include incorporating the third dimension into the model. This would allow processing of multipath signals and would provide a more realistic ocean environment to be modeled.

On the state estimation side, there are many filters that can be used for target tracking, and a detailed study of the feasibility of any of these filters would be beneficial. If a third dimension were added to the TLA model, the filters would need to be updated as well. The output to operators can almost always be improved. With all the information that is available to, and being used by the various filters, it would seem that better methods for data presentation could be developed.

Combining these models would allow the user to close the control loop and look at algorithms that can be used to best track a target. These algorithms do not need to only control the motion but could also be used to generate dynamic decision aids used by submarine or surface ship operators engaging in submerged target tracking.

APPENDIX A. TLA MODEL INITIALIZATION AND PLOTTING SHELL FILES

A. TLA_SIM.M: TLA SIMULATION SHELL

%TLA_Sim initializes and simulates a TLA and plots the output from the %array. Files that are necessary to have in execution path in order to run TLA_Sim include:

```
% -TLA_beam_setup.m
% -TLA.slx
% -Beampattern_DC.m
% -dolph_bw.m
% -dolph.m
% -bwidth.m
% -steer.m
% -ploy2.m
%--These last 4 files can be found on Matlab File exchange on
%      Mathworks website in the "Electromagnetic Waves & Antennas
%      Toolbox" authored by Sophocles Orfanidis.
%      http://www.mathworks.com/matlabcentral/fileexchange/...
%      ...4456-electromagnetic-waves-antennas-toolbox.
%      File: ewa.zip
%-----
---%
%Parameters and initial geometry that need to be specified.
d = 0.4;      %Distance between hydrophone elements in units of
wavelength.
N = 7;        %Number of hydrophone elements in array.
Ra = 30;      %Sidelobe reduction level relative to main lobe in dB.
Base_Freq = 50;      %in hz.
Sim_length = 3600;   %Sec.
Sim_step = .005;     %Sec.
Output_step = 0.5;   %Sec.
Beams_To_Remove = 2;%Number of beams to remove from each endfire.
%Defining Coefficients to shape endfire response.
Acoeff = 1.4;
Bcoeff = 0.5;
Ccoeff = 3.5;
%Initializing the target and array states:
%Enter positions in NM and speeds in knots, the program uses m/s and
meters internally.
Xarray_pos = [0;0];
Xarray_vel = [15;0];
Xtarg_pos = [20,5];
Xtarg_vel = [-20,0];
%Conversion to m/s and meters for units--DO NOT CHANGE FOLLOWING 2
%EQUATIONS
Xao =
[Xarray_pos(1)*1852;Xarray_vel(1)*0.514444;Xarray_pos(2)*1852;Xarray_ve
l(2)*0.514444];
```

```

Xto =
[Xtarg_pos(1)*1852;Xtarg_vel(1)*0.514444;Xtarg_pos(2)*1852;Xtarg_vel(2)
*0.514444];
%***** THERE SHOULD BE NOTHING TO ROUTINELY CHANGE BELOW HERE.
%*****
%-----
%Parameters that will not change often:
ph0 = 90;           %Initial steering angle in degrees (90 is
broadside)
Psi = [ph0];        %Starting beam
Sound_Spd = 1500;    %m/s -Assumes a constant sound speed.
Source_Level = 140;  %Source Level of the Target in dBm. General noise
%                    levels for submarines are: Noisy--140, Quiet--120,
%                    Very Quiet--100 dBm.
Source_Level = 10^(Source_Level/20); %Conv to Amp. of pressure wave
lambda = Sound_Spd/Base_Freq;
%-----

%Generate vectors and arrays necessary for TLA simulation in Simulink
and
%plot the far-field beam pattern of beams to be processed.

[Aw,Awm,Psi,Nspace,Steer_TD,M] =
TLA_beam_setup(d,N,ph0,Ra,lambda,Beams_To_Remove);

%Allow User to ensure no grating lobes.

result = inputdlg('Are you satisfied with Beam pattern, and that it
does not contain grating lobes? (y/n): ','Continue?',1);
result = char(result);
if result == 'n',
    fprintf('Exited because you were not satisfied with Array Setup.')
    return,
else if result == 'y',

else
    fprintf('Was expecting "y or n", exiting.');
```

```

    return,
    end
end
sim('TLA')
plot(Tbrng,time_vect,Brng,time_vect,-Brng,time_vect)

```

B. TLA_BEAM_SETUP.M: BEAM SETUP FUNCTION FILE

```

% TLA_beam_setup.m - Returns Array weights and beam steering angles for
% a set of beams that are spaced at -3dB down point.
%
% Usage: [Aw,Awm,Psi,Nspace,Steer_TD] =
%         TLA_beam_setup(d,N,ph0,Ra,lambda,Beams_To_Remove)
%
% Aw = Normalized vector ( of size(N,1) ) of amplitude weights
% computed by using the Dolph-Chebyshev method for N array elements.
% Awm = Aw in NxN matrix form.
% Psi = Beam steering angles for -3dB spaced main lobes.

```

```

% Nspace = Array used later to generate array spacing.
% Steer_TD = Array steering delay vectors.
% d = element spacing in units of wavelength.
% N = Number of hydrophone array elements.
% ph0 = Center steering angle (normally 90 degrees)
% Ra = The sidelobe reduction level from the main lobe power.
% lambda = Wavelength of base frequency.
% Beams_To_Remove = number of beams to remove to eliminate grating lobes.

% Z.H. Stiles - 2013
function [ Aw,Awm,Psi,Nspace,Steer_TD,M,PSOUT ] = TLA_beam_setup(
d,N,ph0,Ra,lambda,Beams_To_Remove )

if nargin==0, help TLA_beam_setup; return; end

edist = d*lambda; %Distance expressed in units of m/s.
Psi = [ph0]; %Starting beam

%-----
%Setting up nmat a matrix containing the values nmat = [-N'...0...N']'
for the
%odd case and nmat = [-N'...1 1. ..N']' for the even case. N' = N/2 or
%(N-1)/2 for the even or odd case.
nmat = [];
if rem(N,2) == 0,
    Nprime = N/2;

    for kk = 1:Nprime,
        nmat = [nmat; kk];
    end
    nmat = [flipud(nmat);nmat];
else
    Nprime = (N-1)/2;
    for kk = 1:(2*Nprime+1),
        nmat = [nmat;abs(kk-(Nprime+1))];
    end

end

%end the if statement. nmat is done.
%-----

[a,dph] = dolph(d,ph0,N,Ra);
Aw = [abs(a/max(a))]; %Normalized Array weights in column form

%bearings of +/- 3dB down points
k_p = ph0 + dph/2;
k_n = ph0 - dph/2;
%building array of steering angles
Psi = [k_n; Psi; k_p];

while k_p <= 180,
    Psi_next = dolph_bw(d,k_p,N,Ra);
    k_p = k_p + Psi_next/2;

```

```

    k_n = k_n - Psi_next/2;
    Psi = [k_n;Psi;k_p];
    PSOUT = Psi;
end

Psi = [Psi(Beams_To_Remove + 1 : size(Psi,1) - Beams_To_Remove)];
%eliminating beams close to endfires not spaced correctly.
M = size(Psi,1); %number of beams processing

Psi(1,1) = 0.001;
Psi(M,1) = 179.999;

%Plotting the beam pattern
Beampattern_DC(Aw,d,0);
hold on
k = 1;

for k = 2:size((Psi),1)
    Beampattern_DC(Aw,d,Psi(k));
end
hold off

%Plotting of beams complete

Aw = Aw;
%forming Aw in a NxM matrix (repeaing Aw M times)
Awm = zeros((N*M),1);
Awm(1:N,1) = Aw;
for mmm = 2:M,
    Awm(N*mmm-N+1:N*mmm) = [Aw];
end
Awm;

%Setup array used to generate element spacing

Nspace = [linspace(0,N-1,N)]';

%Computing the steering delay vectors that will be used (minus the spd
of sound)

Steer_TD = [];
for mm = 1:M,
    for nn = 1:N,
        Steer_TD = [Steer_TD;-sign(nn-
N/2)*cos(Psi(mm)*pi()/180)*edist*nmat(nn)];
    end
end

Steer_TD = Steer_TD + abs(min(Steer_TD));

end

```

C. BEAMPATTERN_DC.M: PLOTTING FAR-FIELD BEAM PATTERN USING DOLPH-CHEBYSHEV AMPLITUDE AND PHASE WEIGHTING

```
% Beampattern_DC.m - Plots the far-field beam pattern for
%                      Dolph-Chebyshev weighted array consisting of N
%                      elements for a particular steering angle.
%
% Usage: Beampattern_DC(Aw_DC,d,SA)
%
% Aw_DC = Normalized vector ( of size(N,1) ) of amplitude weights
%         computed by using the Dolph-Chebyshev method for N array
elements.
%
% d      = element spacing in units of wavelength.
%
% SA     = Steering angle in degrees for the beam to be plotted.
%
% Z.H. Stiles - 2013

function [] = Beampattern_DC(Aw_DC,d,SA)

if nargin==0, help Beampattern_DC; return; end

Psi = 0:0.01:2*pi; %plotting points
L = size((Aw_DC),1);
P = size(Psi,2);
SAr = SA*pi()/180;

if rem(L,2) == 0
    %even case
    k = L/2;
    Anew = Aw_DC(k+1:L,1);

    Sup = 0;
    Sdn = 2*sum(Anew);
    Up = 0;
    Dn = 0;

    for n = 1:k,
        Up = 2*Anew(n,1)*cos(2*pi()*(n-0.5)*d*(cos(Psi)-cos(SAr)));
        Sup = Sup + Up;
        Up = 0;
    end
    %Normalized array factor
    Sn = abs(Sup./Sdn);
    Sn_dB = 40 + 20*log10(Sn);
    for b = 1:P,
        if Sn_dB(b) < 0,
            Sn_dB(b) = 0;
        end
    end %End even Case
else
    %odd case
    k = (L-1)/2;
```

```

Anew = Aw_DC(k+2:L,1);
Ao = Aw_DC(k+1,1);

Sup = Ao*ones(1,P);
Sdn = Ao*ones(1,P);
Up = 0;
Dn = 0;

for n = 1:k,
    Up = 2*Anew(n,1)*cos(2*pi()*n*d*(cos(Psi)-cos(SAr)));
    Sup = Sup + Up;
    Up = 0;
    Dn = 2*Anew(n,1);
    Sdn = Sdn + Dn;
    Dn = 0;
end
%Normalized array factor
Sn = abs(Sup./Sdn);
Sn_dB = 40 + 20*log10(Sn);
for b = 1:P,
    if Sn_dB(b) < 0,
        Sn_dB(b) = 0;
    end
end %End Odd Case
end

polar(Psi,Sn_dB);

end

```

D. DOLPH_BW.M: CALCULATING BEAMWIDTHS

This file has been edited file from [17].

```

% dolph_bw.m - Dolph-Chebyshev beam width
%
% Usage: [Bw] = dolph(ph0, N, R)
%
% ph0 = beam angle in degrees (broadside ph0=90)
% N   = number of array elements (even or odd)
% R   = relative sidelobe level in dB, (e.g. R = 30)
%
% Bw = 3-dB beamwidth in degrees
%
% FROM:
% S. J. Orfanidis - 1997 - www.ece.rutgers.edu/~orfanidi/ewa
% Coded original dolph.m function. This is edited to just return beam
% widths.
%
% Z. H. Stiles - 2013 (Edit only)

function [Bw] = dolph_bw(d,ph0, N, R)

if nargin==0, help dolph; return; end

```



```

N1 = N - 1; % number of pattern zeros
Ra = 10^(R/20); % sidelobe level in absolute units
x0 = cosh(acosh(Ra)/N1); % scaling factor
x3 = cosh(acosh(Ra/sqrt(2))/N1); % 3-dB Chebyshev variable x
psi3 = 2*acos(x3/x0); % exact 3-dB frequency
dps = 2*psi3; % 3-dB width

Bw = bwidth(d, ph0, dps); % 3-dB width in phi-space

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TLA SIMULINK FILES

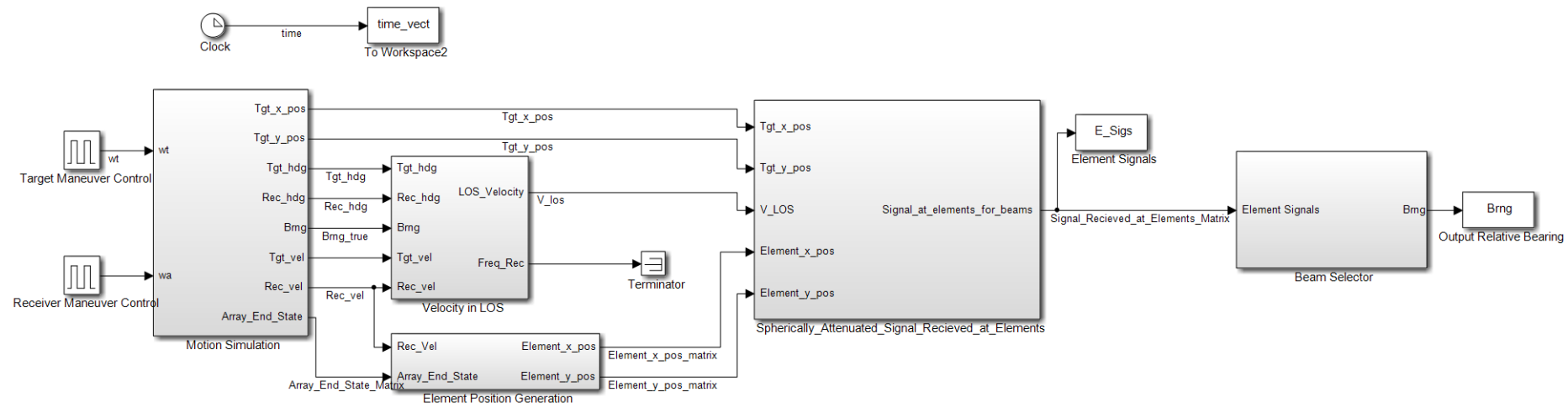


Figure 38. TLA.slx Simulink top level view.

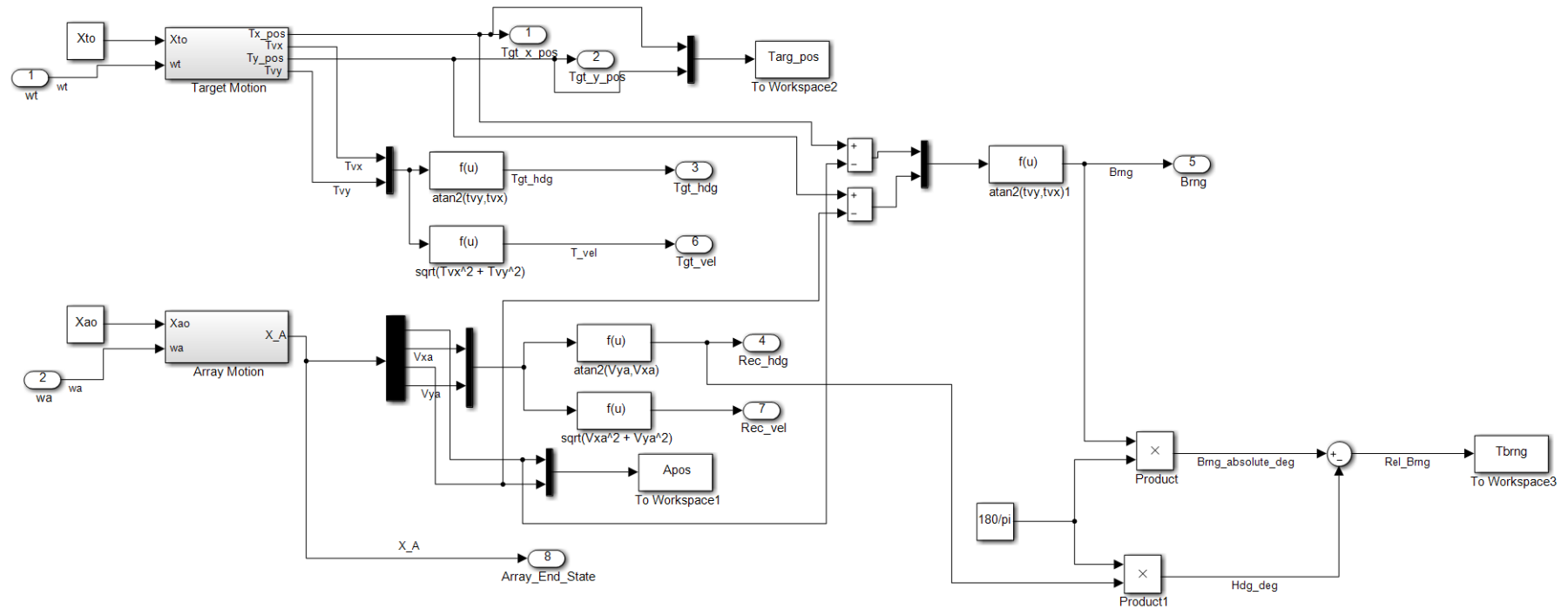


Figure 39. Interior of motion simulation subsystem (TLA/Motion Simulation).

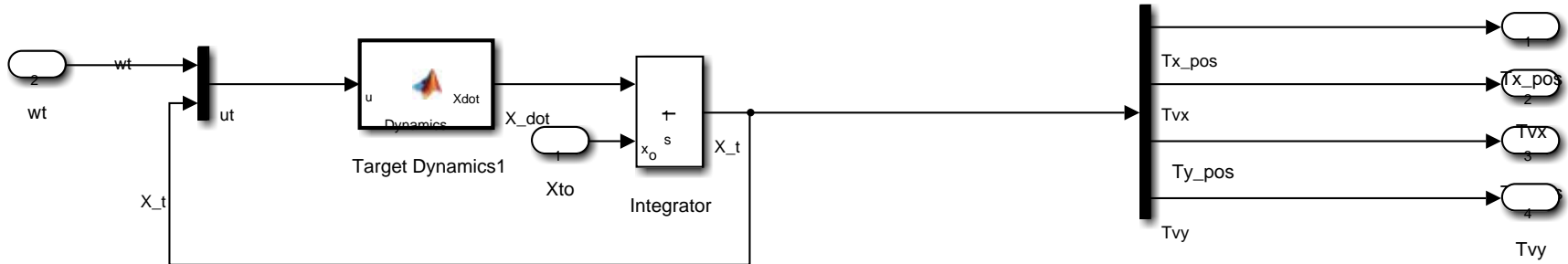


Figure 40. Interior of target motion subsystem block (TLA/Motion Simulation/Target Motion).

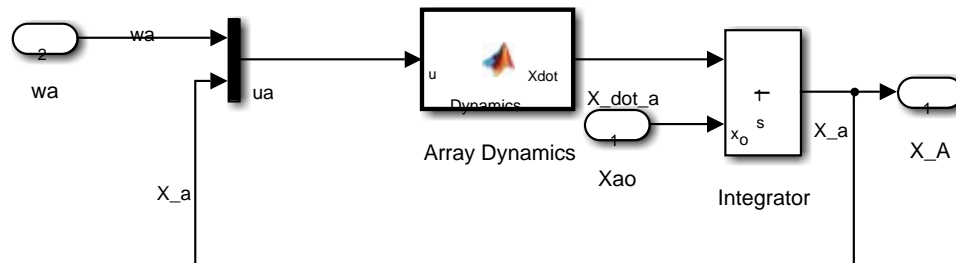


Figure 41. Interior of array motion subsystem block (TLA/Motion Simulation/Array Motion).

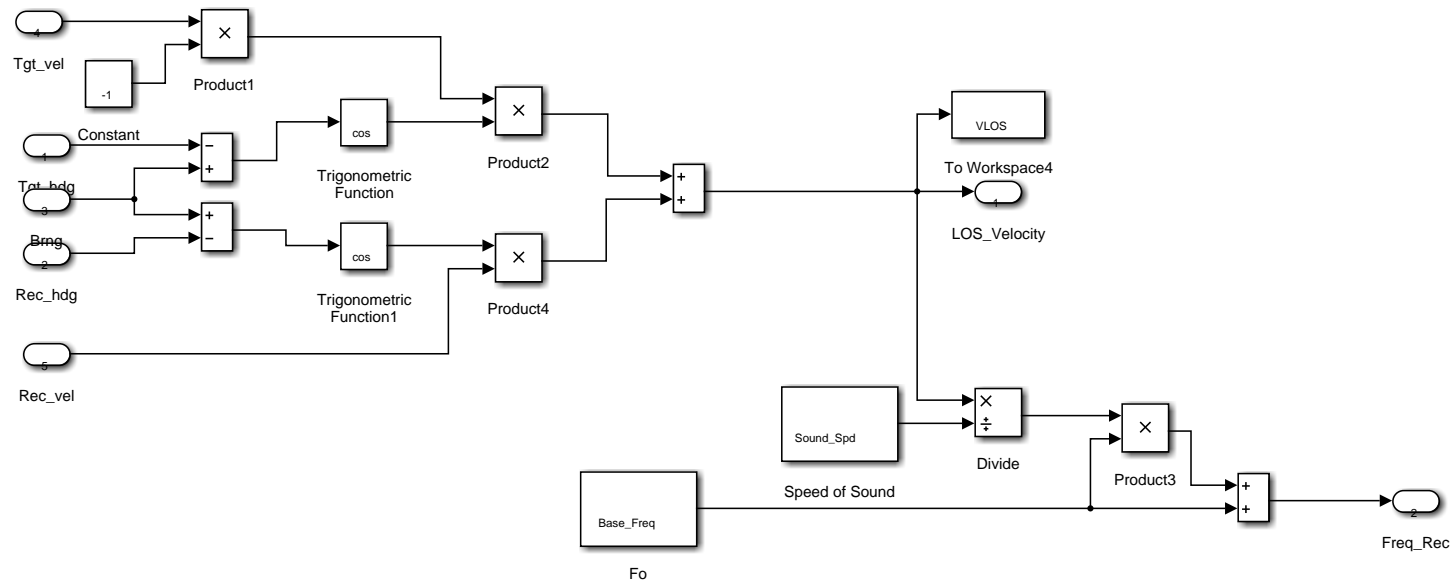


Figure 42. Interior of velocity in LOS subsystem block (TLA/Velocity in LOS).

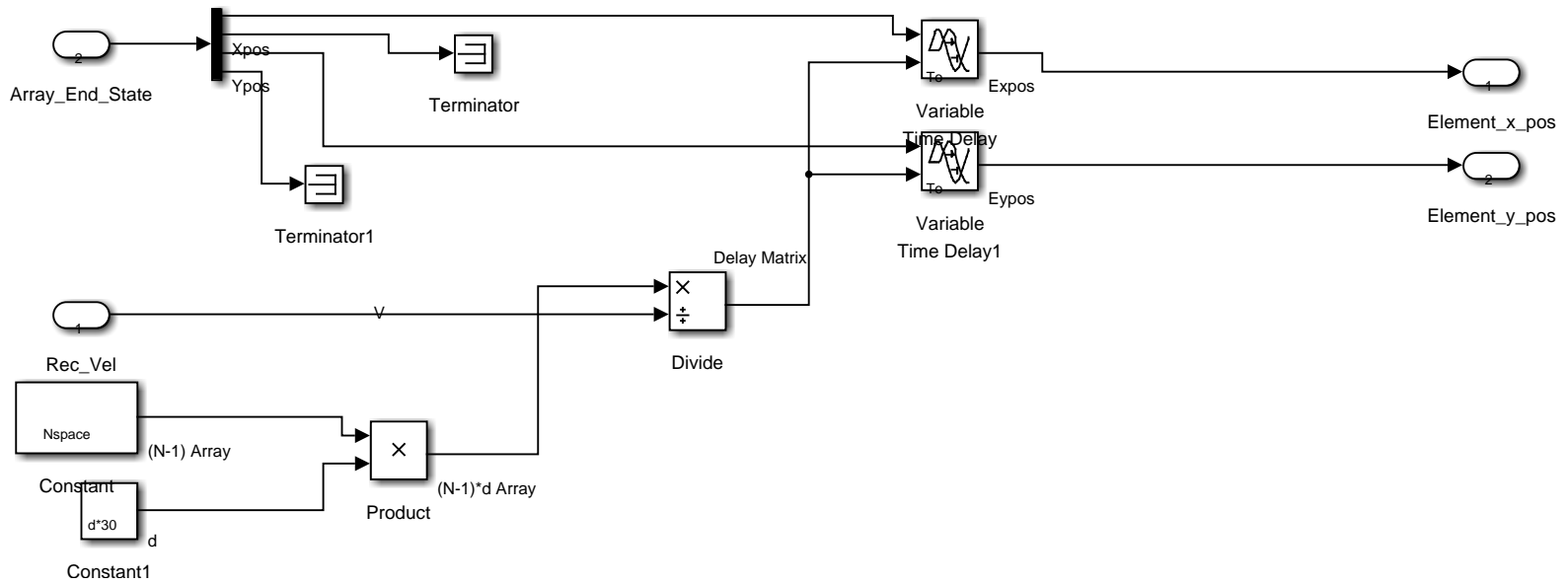


Figure 43. Interior of the element position simulation subsystem block (TLA/Element Position Simulation).

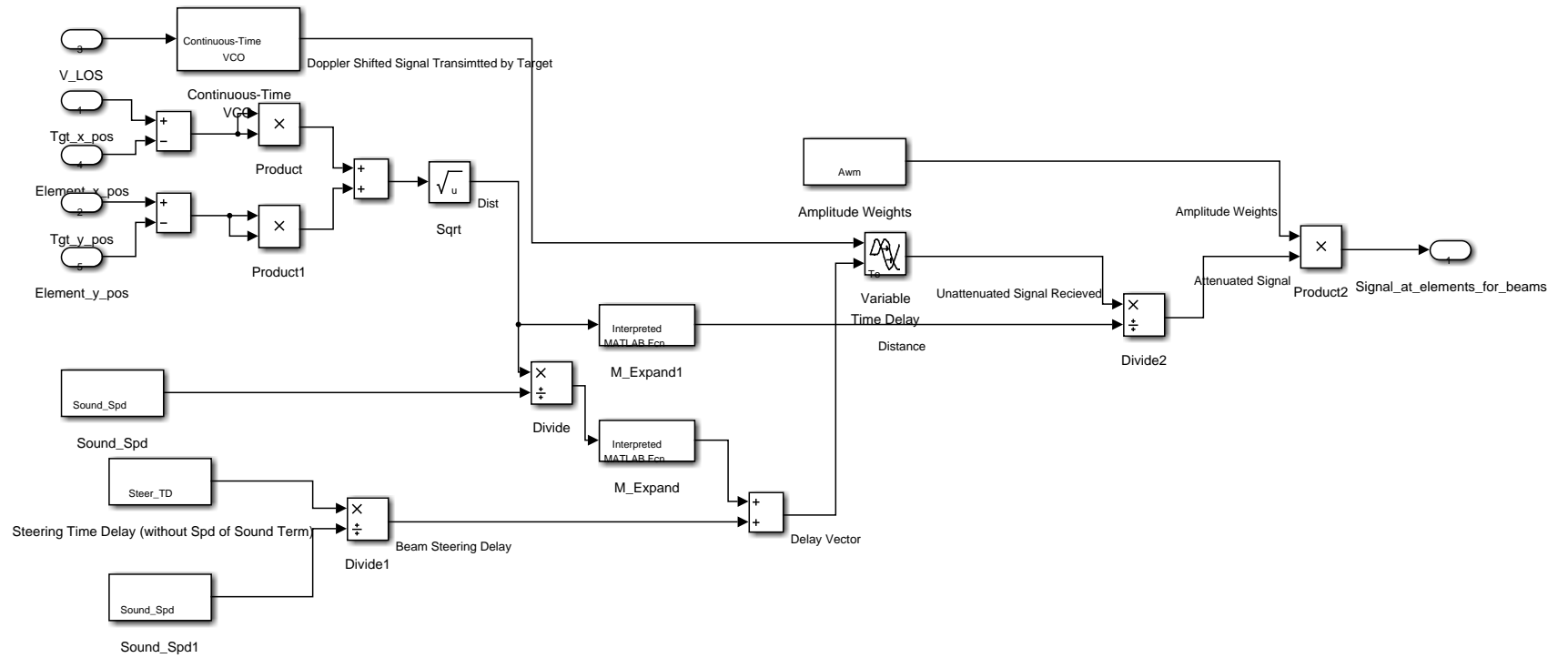


Figure 44. Interior of spherically attenuated signal received at elements subsystem block (TLA/SASRaES).

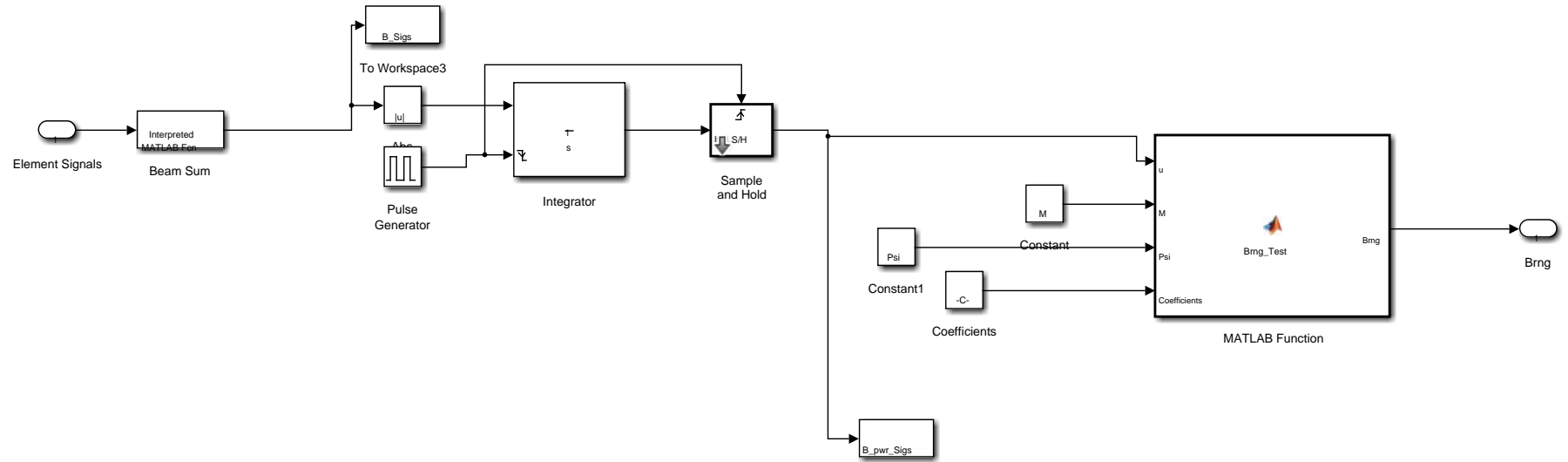


Figure 45. Interior of beam selector subsystem (TLA/Beam Selector).

A. EMBEDDED MATLAB FUNCTION BLOCK “DYNAMICS”

```
function Xdot = Dynamics( u )
%Used for Motion Simulation.
w = u(1);
x = u(2);
Vx = u(3);
y = u(4);
Vy = u(5);
Xdot = [Vx;-w*Vy;Vy;w*Vx];
end
```

B. M_EXPAND.M

```
function [ Expanded_Array ] = M_Expand(u)
%Expands an input array of size nx1 to (n*m)x1.

M = u(1,1);
ArrayIn = u(2:size(u,1),1);
aa = 2;
Expanded_Array = zeros(M*(size(ArrayIn,1)),1);
Expanded_Array(1:size(ArrayIn,1),1) = ArrayIn;
while aa <= M ,
    Expanded_Array(((aa - 1)*size(ArrayIn,1)+1):aa*size(ArrayIn,1),1) =
    [ArrayIn];
    aa = aa + 1;
end
end
```

C. BEAM_SUM.M

```
function [ Beam_out ] = Beam_sum( u )
%This function will form the beams by adding the delayed and weighted
%signals for each beam up. need input u = [N;M;input];

N = u(1);
M = u(2);
signals = u(3:size(u,1));

Beam_out = zeros(M,1);
for jj = 1:M,
    Beam_out(jj) = sum(signals(N*jj-(N-1):N*jj));
end
end
```

D. EMBEDDED MATLAB FUNCTION BLOCK “BRNG_TEST”

```
function Brng = Brng_Test( u, M, Psi, Coefficients)
%Reports the max input port number.
Bsig = u;
Mtemp = M;
A = Coefficients(1);
B = Coefficients(2);
```

```

C = Coefficients(3);

[Max_val, Max_Beam] = max(Bsig(:,1));

if Max_val == 0,
    Brng = 0;

else if Max_Beam ~= 1 && Max_Beam ~= 2 && Max_Beam ~= Mtemp && Max_Beam
    ~= Mtemp-1,

    L1 = Max_Beam + 1;
    R1 = Max_Beam - 1;
    Brng = 0;

    if Bsig(L1) > Bsig(R1),

        Max_deltaB = abs(Psi(L1) - Psi(Max_Beam))/2;
        Ratio = Bsig(Max_Beam)/Bsig(L1);
        Ratio = min(sqrt(2),Ratio);
        Brng = Psi(Max_Beam,1) + Max_deltaB*(1-((Ratio-1)/(sqrt(2)-
1)))));

    else if Bsig(R1) >= Bsig(L1),

        Max_deltaB = abs(Psi(Max_Beam) - Psi(R1))/2;
        Ratio = Bsig(Max_Beam)/Bsig(R1);
        Ratio = min(sqrt(2),Ratio);

        Brng = Psi(Max_Beam,1) - Max_deltaB*(1-((Ratio-
1)/(sqrt(2)-1)))));

    end
end
else if Max_Beam == 1,

    L1 = Max_Beam + 1;
    Delta_B = Psi(2)/2;
    EF_Ratio = Bsig(1)/Bsig(2);
    Brng = A*(Delta_B) - A*(Delta_B)*min(1,(EF_Ratio-1)/(C-1));

    else if Max_Beam == 2 && Bsig(1)/Bsig(3) >= 1,
        Delta_B = Psi(2)/2;
        EF_Ratio = Bsig(1)/Bsig(2);
        Brng = A*Delta_B + 0.5*(2-A)*Delta_B *(1-
max(0,min(1,(EF_Ratio-B)/(1-B)))));

    else if Max_Beam == 2 && Bsig(1)/Bsig(3) < 1,

        L1 = Max_Beam+1;
        Max_deltaB = abs(Psi(L1) - Psi(Max_Beam))/2;

```


APPENDIX C. FILTER_SIM: STATE ESTIMATION INITIALIZATION AND PLOTTING SHELL

```
%Filter_Sim.m This file initalizes and simulates a target tracking  
%encounter simulation and uses TLA outputs (generated in simulation) to  
%drive EKF's used for State Estimation.
```

```
%LT Zack Stiles July 2013
```

```
%Setput Starting Geometry:
```

```
Xarray_pos = [0;0];           %X;Y position in KM.  
Xarray_vel = [9;2];          %Vx;Vy in Knots  
Xtarg_pos = [10,10];          %X;Y position in KM.  
Xtarg_vel = [9,2];            %Vx;Vy in Knots
```

```
%Simulation Parameter Setup:
```

```
dlt = 1;                      %Step time in sec  
NRuns = 5;                    %Number of simulations  
Stop_Time = 90000;            %Simulation length in sec
```

```
%Parameter Setup
```

```
Sound_Spd = 1500;             %m/s  
Base_Freq = 55;               %Target Base Freq in Hz.  
lambda = Sound_Spd/Base_Freq;
```

```
%Filter coefficient Setup:
```

```
q = (2*pi/90)^2;              %Position and Velocity Variation due to  
plant noise  
qf = 0;                       %Base Freq variation due to plant noise  
Pint = [3750^2 0 0 0 0;0 1.6541 0 0 0; 0 0 3750^2 0 0; 0 0 0 1.6541 0;  
0 0 0 0 3.2673e-7];  
PintBO = [50 0 0 0;0 1 0 0; 0 0 50 0; 0 0 0 1 ];  
Brng_Var = 3.04617e-4;  
Freq_Var = 4.5e-7;
```

```
%Source Level of the Target dBm : Noisy is 140, Quiet is 120, Very  
Quiet 100  
%dBm  
Source_Level = 140;
```

```
%Conversion to m/s and meters for units
```

```
*****Do Not alter this section*****
```

```

Xao =
[Xarray_pos(1)*1000;Xarray_vel(1)*0.514444;Xarray_pos(2)*1000;Xarray_ve
l(2)*0.514444];
Xto =
[Xtarg_pos(1)*1000;Xtarg_vel(1)*0.514444;Xtarg_pos(2)*1000;Xtarg_vel(2)
*0.514444];
%Starting estimate
Xhatint = [Xto;Base_Freq]-[100; 1 ; 100; 1; .002];
XhatHint = [Xto;Base_Freq];
XhatBOint = Xto;
%*****
%*****
%Data collection and plotting:
Tout = zeros(Stop_Time/dlt + 1,2);
Xhat_Out = zeros(Stop_Time/dlt + 1,5);
XhatH_Out = zeros(Stop_Time/dlt + 1,5);
XhatBO_Out = zeros(Stop_Time/dlt + 1,4);
Error_Out = zeros(Stop_Time/dlt + 1,8);

for jj = 1:NRuns,

    sim('Tracking_Sim');

    if jj == 1,
        Tout = Tout + Tpos;
    end

    Xhat_Out    = Xhat_Out    + Xhat;
    XhatH_Out   = XhatH_Out   + XhatH;
    XhatBO_Out  = XhatBO_Out  + XhatBO;

    E_pos = sqrt((Tpos(:,1)-Xhat(:,1)).^2 + (Tpos(:,2)-
Xhat(:,3)).^2);
    E_posH = sqrt((Tpos(:,1)-XhatH(:,1)).^2 + (Tpos(:,2)-
XhatH(:,3)).^2);
    E_posBO = sqrt((Tpos(:,1)-XhatBO(:,1)).^2 + (Tpos(:,2)-
XhatBO(:,3)).^2);
    E_Bng = abs(atan2(Tpos(:,2)-Rpos(:,2),Tpos(:,1)-Rpos(:,1)) -
zhat(:,1));
    E_BngH = abs(atan2(Tpos(:,2)-Rpos(:,2),Tpos(:,1)-Rpos(:,1)) -
zhatH(:,1));
    E_BngBO = abs(atan2(Tpos(:,2)-Rpos(:,2),Tpos(:,1)-Rpos(:,1)) -
zhatBO(:,1));
    E_BF = Base_Freq - Xhat(:,5);
    E_BFH = Base_Freq - XhatH(:,5);
    Emat = [E_pos,E_posH,E_posBO,E_Bng,E_BngH,E_BngBO,E_BF,E_BFH];
    Error_Out = Error_Out + Emat;

end

Xhat_Mean = Xhat_Out./NRuns;
XhatH_Mean = XhatH_Out./NRuns;
XhatBO_Mean = XhatBO_Out./NRuns;
Mean_Error = Error_Out./NRuns;

```

```

figure(1)
plot(Tpos(:,1),Tpos(:,2),Xhat_Mean(:,1),Xhat_Mean(:,3),XhatH_Mean(:,1),
XhatH_Mean(:,3),XhatBO_Mean(:,1),XhatBO_Mean(:,3),Rpos(:,1),Rpos(:,2));
axis equal;
title('True Target Trajectory and EKF Mean Measurement Trajectories')
legend('True Trajectory','Mean Measurement EKF','Mean Measurement EKF-
H','Mean Measurement EKF-BO','Receiver Trajectory')
xlabel('x (Meters)')
ylabel('y (Meters)')

%Plot of mean positional errors
figure(2)
subplot(2,1,1)
plot(time_vect,Mean_Error(:,1),time_vect,Mean_Error(:,2),time_vect,Mean
_Error(:,3));
title('Mean Positional Errors')
legend('EKF Error','EKF-H Error','EKF-BO Error')
xlabel('Time (sec)')
ylabel('Range (Meters)')
hold on

subplot(2,1,2)
plot(time_vect,sqrt((RecState(1,:)-TgtState(1,:)).^2+(RecState(3,:)-
TgtState(3,:)).^2));
title('Range vs. Time')
legend('Tgt Range')
xlabel('Time (sec)')
ylabel('Range (m)')
hold off

figure(3)
subplot(2,1,1)
%Plot of mean Brng Errors
plot(time_vect,Mean_Error(:,4)*180/pi,time_vect,Mean_Error(:,5)*180/pi,
time_vect,Mean_Error(:,6)*180/pi);
title('Estimated Bearing Errors')
legend('EKF Bearing Error','EKF-H Bearing Error','EKF-BO Bearing
Error')
xlabel('Time (sec)')
ylabel('Bearing Error (deg)')
hold on

subplot(2,1,2)
plot(time_vect,Meas(:,1)*180/pi(),time_vect,zhat(:,1)*180/pi,time_vect,
zhatH(:,1)*180/pi,time_vect,zhatBO(:,1)*180/pi)
title('Measured and estimated target Bearing')
legend('Measured Bearing','EKF Mean Bearing Estimate','EKF-H Mean
Bearing Est','EKF-BO Mean Bearing Est')
xlabel('Time (sec)')
ylabel('Bearing (deg)')

figure(4)

```

```
%Plot of mean BF errors
plot(time_vect,Mean_Error(:,7),time_vect,Mean_Error(:,8));
title('Base Frequency Error')
legend('EKF Base Freq Error','EKF-H Base Freq Error')
xlabel('Time (sec)')
ylabel('Base Frequency Error (Hz)')
```


APPENDIX D. TRACKING_SIM.SLX: STATE ESTIMATION SIMULINK FILE

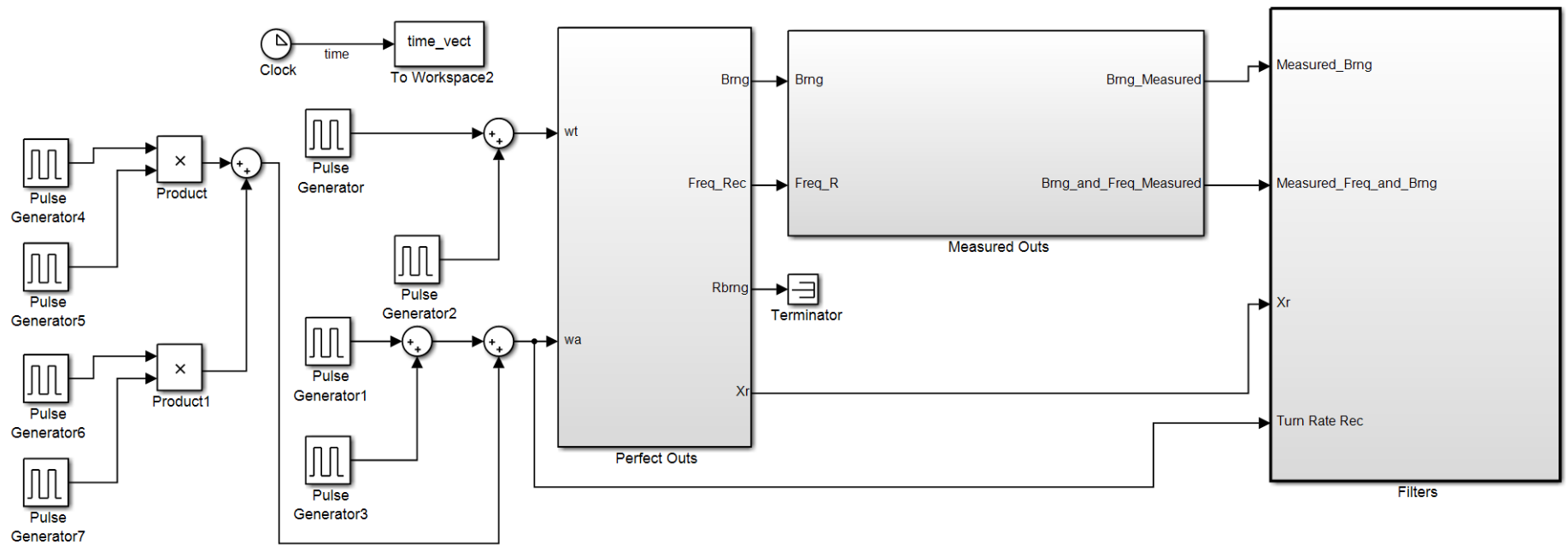


Figure 46. Top level view of Tracking_Sim.slx.

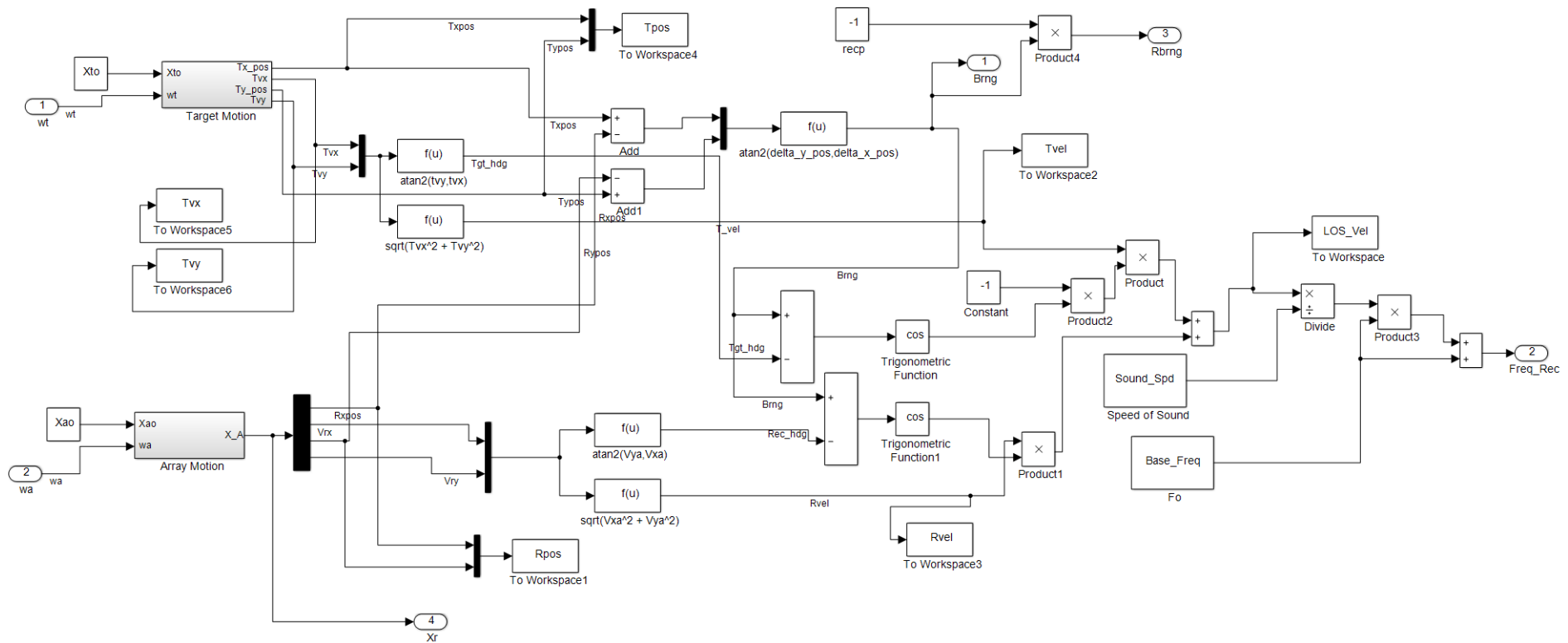


Figure 47. Interior of perfect outs subsystem block (Tracking_Sim/Perfect Outs).

Note: Target and Array Motion Subsystems have the same interior contents as the Target Motion and Array Motion Subsystems in the TLA.slx model.

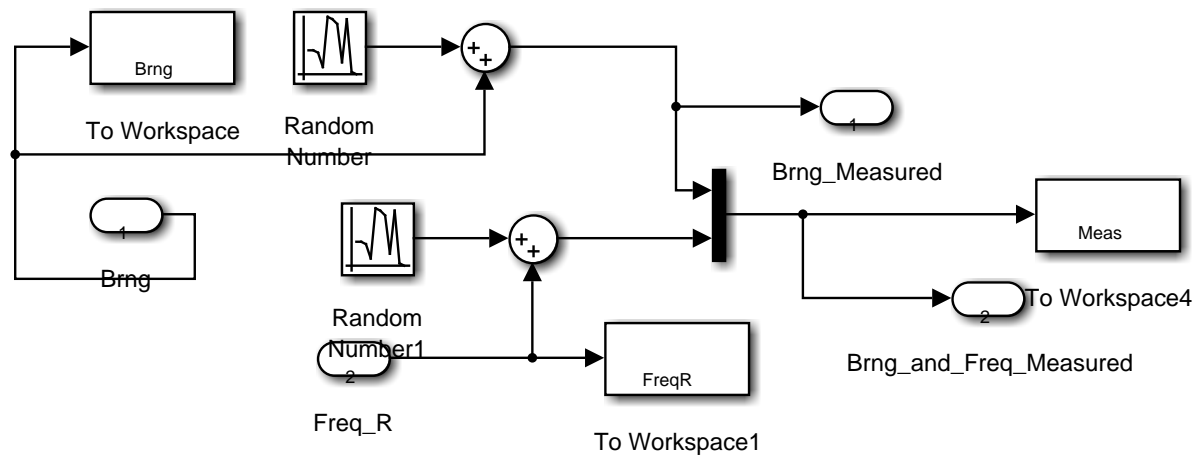


Figure 48. Interior of measured outs subsystem block (Tracking_Sim/Measured Outs).

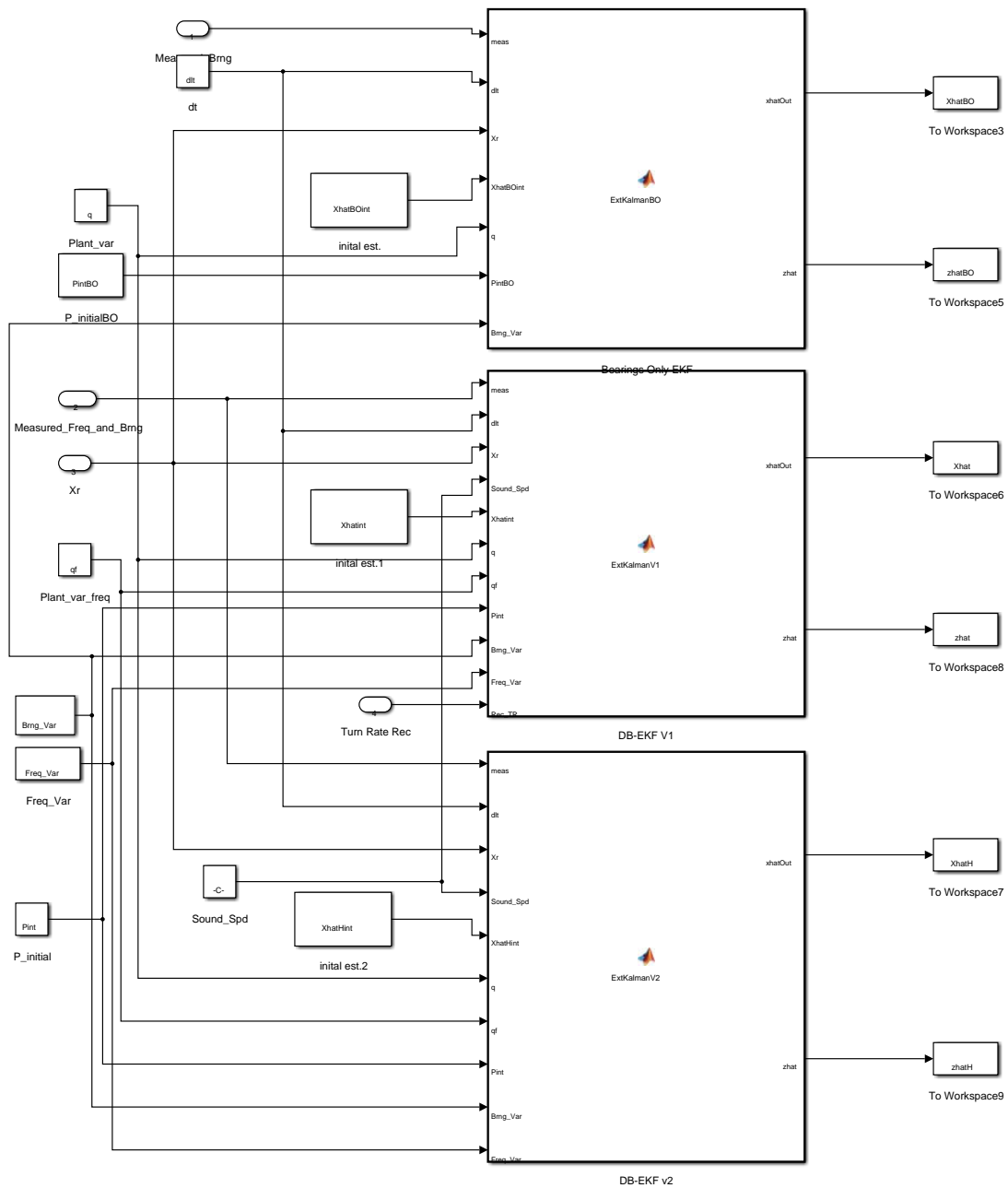


Figure 49. Interior of filter subsystem block (Tracking_Sim/Filter).

A. EMBEDDED MATLAB FUNCTION BLOCK “EKFKALMANBO”

```

function [xhatOut,zhat] = ExtKalmanBO(meas, dlt,
Xr,XhatBOint,q,PintBO,Brng_Var)
% Bearing Only

persistent P xhat Q R
if isempty(P)
    % First time through the code so do some initialization

    xhat = XhatBOint;
    P = PintBO;
    Q = q*[dlt^3/3 dlt^2/2 0 0; dlt^2/2 dlt 0 0; 0 0 dlt^3/3 dlt^2/2; 0
0 dlt^2/2 dlt] ;
    R = diag([Brng_Var]);

end

% Calculate the Jacobians for the state and measurement equations
F = [1 dlt 0 0;0 1 0 0;0 0 1 dlt;0 0 0 1];

H1 = (Xr(3)-xhat(3))/((xhat(1)-Xr(1))^2 + (xhat(3)-Xr(3))^2);
H3 = (xhat(1)-Xr(1))/((xhat(1)-Xr(1))^2 + (xhat(3)-Xr(3))^2);

H = [H1,0,H3,0];

zhat = atan2((xhat(3)-Xr(3)),(xhat(1)-Xr(1)));

% Propagate the state and covariance matrices

P = F*P*F' + Q;

% Calculate the Kalman gain
K = P*H'*inv(H*P*H' + R);

% Update the state and covariance estimates
xhat = xhat + K*(meas - zhat);
P = (eye(size(K,1))-K*H)*P*(eye(size(K,1))-K*H)' + K*R*K';

%Limiting the possible velocity output:
if abs(xhat(2)) > 15*0.514444,
    xhat(2) = sign(xhat(2))*15*0.51444;
end
if abs(xhat(4)) > 15*0.51444,
    xhat(4) = sign(xhat(4))*15*0.51444;
end

% Post the results
xhatOut = xhat;
xhat = F*xhat;

```

B. EMBEDDED MATLAB FUNCTION BLOCK “EXTKALMANV1” (DB-EKF V1)

```
function [xhatOut,zhat] = ExtKalmanV1(meas, dlt, Xr, Sound_Spd,
Xhatint,q,qf,Pint,Brng_Var,Freq_Var,Rec_TR)
% This Embedded MATLAB Function implements an extended Kalman filter.
%
% The states of the process are given by
% x = [x_position; x_velocity; y_position; y_velocity; freq_base];
%
% and the measurements are given by
% h = [bearing;freq_received]
%
% where
% brng = atan2(y_position/x_position)
% freq_received = freq_base(1 + Vlos/c)
% Vlos = deltaVx*cos(brng) + deltaVy*sin(brng)

% Define storage for the variables that need to persist
% between time periods.

persistent P xhat Q R R_Large
if isempty(P)
    % First time through the code so do some initialization
    xhat = Xhatint;
    P = Pint;
    Q = q*[dlt^3/3 dlt^2/2 0 0 0; dlt^2/2 dlt 0 0 0; 0 0 dlt^3/3
dlt^2/2 0; 0 0 dlt^2/2 dlt 0; 0 0 0 0 dlt*qf/q] ;
    R = diag([Brng_Var;Freq_Var]);
    R_Large = diag([1e6;1e6]);
end

% Calculate the Jacobians for the state and measurement equations
F = [1 dlt 0 0 0 ; 0 1 0 0 0; 0 0 1 dlt 0; 0 0 0 1 0; 0 0 0 0 1];

deltaVx = Xr(2) - xhat(2);
deltaVy = Xr(4) - xhat(4);
deltaX = xhat(1) - Xr(1);
deltaY = xhat(3) - Xr(3);
brngHat = atan2((deltaY),(deltaX));
VLOS_Hat = (deltaVx)*cos(brngHat) + (deltaVy)*sin(brngHat);
freq_recHat = xhat(5)*(1 + VLOS_Hat/Sound_Spd);
zhat = [brngHat;freq_recHat];

partial_brng_x1 = (-deltaY) / ((deltaY)^2 + (deltaX)^2);
partial_brng_x3 = (deltaX) / ((deltaY)^2 + (deltaX)^2);

H11 = partial_brng_x1;
H13 = partial_brng_x3;

T1 = (xhat(5)/Sound_Spd)*partial_brng_x1;
T2 = (deltaVy)*cos(brngHat);
T3 = (deltaVx)*sin(brngHat);
H21 = T1*(T2-T3);
```

```

H22 = -(xhat(5)/Sound_Spd)*cos(brngHat);

T4 = (xhat(5)/Sound_Spd)*partial_brng_x3;
T5 = (deltaVy)*cos(brngHat);
T6 = (deltaVx)*sin(brngHat);

H23 = T4*(T5-T6);

H24 = -(xhat(5)/Sound_Spd)*sin(brngHat);

H25 = (1+VLOS_Hat/Sound_Spd);

H = [H11,0,H13,0,0; H21,H22,H23,H24,H25];

% Propagate the state and covariance matrices
P = F*P*F' + Q;

% Calculate the Kalman gain
if Rec_TR <= 0.5/pi,

    K = P*H'*inv(H*P*H' + R);

    % Update the state and covariance estimates
    xhat = xhat + K*(meas-zhat);
    P = (eye(size(K,1))-K*H)*P*(eye(size(K,1))-K*H)' + K*R*K';
else

K = P*H'*inv(H*P*H' + R_Large);

% Update the state and covariance estimates
xhat = xhat + K*(meas-zhat);
P = (eye(size(K,1))-K*H)*P*(eye(size(K,1))-K*H)' + K*R_Large*K';
end

% Post the results
xhatOut = xhat;

xhat = F*xhat;

```

C. EMBEDDED MATLAB FUNCTION BLOCK “EXTKALMANV2”

```

function [xhatOut,zhat] = ExtKalmanV2(meas, dlt, Xr,
Sound_Spd,XhatHint,q,qf,Pint,Brng_Var,Freq_Var)
% This Embedded MATLAB Function implements an extended Kalman filter.
%
% The states of the process are given by
% x = [x_position; x_velocity; y_position; y_velocity; freq_base];
%
% and the measurements are given by
% h = [bearing;freq_received]
%
% where
% brng = atan2(y_position/x_position)

```

```

% freq_received = freq_base(1 + Vlos/c)
% Vlos = deltaVx*cos(brng) + deltaVy*sin(brng)

% Define storage for the variables that need to persist
% between time periods.

persistent P xhat Q R
if isempty(P)
    % First time through the code so do some initialization
    xhat = XhatHint;
    P = Pint;
    Q = q*[dlt^3/3 dlt^2/2 0 0 0; dlt^2/2 dlt 0 0 0; 0 0 dlt^3/3
dlt^2/2 0; 0 0 dlt^2/2 dlt 0; 0 0 0 0 dlt*qf/q] ;
    R = diag([Brng_Var;Freq_Var]);
end

% Calculate the Jacobians for the state and measurement equations
F = [1 dlt 0 0 0 ; 0 1 0 0 0; 0 0 1 dlt 0; 0 0 0 1 0; 0 0 0 0 1];

deltaVx = -(xhat(2) - Xr(2));
deltaVy = -(xhat(4) - Xr(4));
deltaX = xhat(1) - Xr(1);
deltaY = xhat(3) - Xr(3);
brngHat = atan2((deltaY),(deltaX));

Rng = sqrt(deltaY^2+deltaX^2);
VLOS_Hat = ((deltaX)*(deltaVx) + (deltaVy)*(deltaY))/Rng;
freq_recHat = xhat(5)*(1 + VLOS_Hat/Sound_Spd);
zhat = [brngHat;freq_recHat];

H11 = -deltaY/Rng^2;
H13 = deltaX/Rng^2;

Mult = xhat(5)/Sound_Spd;

T1 = Mult*deltaY/Rng^3;
T2 = ((deltaVx)*(deltaY)-(deltaVy)*(deltaX));
H21 = T1*T2;

H22 = -Mult*deltaX/Rng;

T3 = Mult*deltaX/Rng^3;
T4 = ((deltaVy)*(deltaX)-(deltaVx)*(deltaY));
H23 = T3*T4;

H24 = -Mult*deltaY/Rng;

H25 = (1+VLOS_Hat/Sound_Spd);
H = [H11,0,H13,0,0; H21,H22,H23,H24,H25];

```



```

%nonupdated = xhat;
P = F*P*F' + Q;

% Calculate the Kalman gain
K = P*H'*inv(H*P*H' + R);

% Update the state and covariance estimates
xhat = xhat + K*(meas-zhat);
P = (eye(size(K,1))-K*H)*P*(eye(size(K,1))-K*H)' + K*R*K';

% Post the results
xhatOut = xhat;
xhat = F*xhat;

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] M. J. Conner. Sustaining undersea dominance. *Proceedings Magazine*, vol. 139/6/1,324. [Online]. Available: <http://www.usni.org/magazines/proceedings/2013-06/sustaining-undersea-dominance>, June 2013.
- [2] Task Force ASW. *Anti-submarine warfare: Concepts of operations for the 21st century*. [Online]. Available: <http://www.navy.mil/navydata/policy/asw/asw-conops.pdf>, [Accessed Aug. 2, 2013].
- [3] R. O'Rourke, "Navy Virginia (SSN-774) Class attack submarine procurement: background and issues for Congress," Congressional Research Service, Washington, DC, CRS Report No. RL32418, June 28, 2013.
- [4] "China sub stalked U.S. fleet," *Washington Times*. [Online]. Available: <http://www.washingtontimes.com/news/2006/nov/13/20061113-121539-3317r/?page=all>, Nov. 13, 2006.
- [5] J. Pomfret and B. Harden, "South Korea to officially blame North Korea for March torpedo attack on warship," *Washington Post*, [Online]. Available: <http://www.washingtonpost.com/wp-dyn/content/article/2010/05/18/AR2010051803094.html>, May 19, 2013.
- [6] S. G. Lemon, "Towed-array history, 1917–2003," *IEEE J. of Ocean. Eng.*, vol. 29, pp. 365–373, Apr. 2004.
- [7] M. S. Cohen and R. V. Brown, "Decision support for attack submarine commanders," Decision Sci. Consortium, Falls Church, VA, Tech. Rep. 80–11, Oct. 1980.
- [8] Robert C. Harney, "Propagation of acoustic radiation," in *Combat Systems, vol. I, part I*, pp. 195–228. Jan. 2004.
- [9] M. Lansky, R. D. Doolittle, B. D. Simmons and S. G. Lemon, "Recent progress in towed hydrophone array research," *IEEE J. of Ocean. Eng.*, vol. 29, pp. 374–387, Apr. 2004.
- [10] B. L. Scala and M. Morelande, "An analysis of the single sensor bearings-only tracking problem," in *Proc. ISIF Intern. Conf. On Information Fusion (FUSION)*, pp. 525–530, 2008.
- [11] K. Kaouri, "Left-right ambiguity resolution of a towed array sonar," M.S. thesis, Math. Institution, Oxford Univ., Oxford, United Kingdom, 2000.

- [12] M.J. Hamilton, S. Kemna and D. Hughes, "Antisubmarine warfare applications for autonomous underwater vehicles: The GLINT09 sea trial results," *J. Field Robotics*, vol. 27, pp. 890–902, Aug. 24, 2010.
- [13] L. J. Ziomek, NPS EC 4450 class notes, Naval Postgraduate School, unpublished. Feb. 2013.
- [14] G. Moernaut and D. Orban. *The basics of antenna arrays* [Online]. Available: http://www.orbanmicrowave.com/The_Basics_of_Antenna_Arrays.pdf
- [15] S. F. Maharimi et al., "Impact of spacing and number of elements on array factor," in *Proc. of Progress in Electromagnetics Research Symp.*, pp. 1550–1553, Kuala Lumpur, Mar. 27–30, 2012.
- [16] D. H. Johnson and D. E. Dudgeon, "Beamforming," in *Array Signal Processing*, Upper Saddle River, NJ: Prentice-Hall, pp. 111–190, 1993.
- [17] S. Orfanidis. *Electromagnetic waves and antennas toolbox* [Online]. Available FTP: <http://www.mathworks.com/matlabcentral/fileexchange/4456-electromagnetic-waves-antennas-toolbox>, File: ewa.zip.
- [18] "MATLAB function," *Mathworks Documentation Center* [Online]. Available: <http://www.mathworks.com/help/simulink/slref/matlabfunction.html>
- [19] Y. Bar-Shalom et al., *Estimation with Applications to Tracking and Navigation*, New York, NY: Wiley-Interscience, p. 200–217, 2001.
- [20] Y. Bar-Shalom et al., *Estimation with Applications to Tracking and Navigation*, New York, NY: Wiley-Interscience, pp. 381–395, 2001.
- [21] G. Welch and G. Bishop, "An introduction to the Kalman Filter," Univ. of North Carolina, Chapel Hill, NC, Tech. Rep. 95-041, July 2006.
- [22] R. G. Hutchins, "Optimal estimation, Kalman filters, and target tracking," Naval Postgraduate School, unpublished. 1997.
- [23] K. Ho and Y. Chan, "Geometric-polar tracking from bearings-only and Doppler-bearing measurements," *Signal Process., IEEE Trans.*, vol. 56, no. 11, pp. 5540–5554, Nov. 2008.
- [24] Y. Bar-Shalom et al, "Linear dynamic systems with random inputs," in *Estimation with Applications to Tracking and Navigation*, New York, NY: Wiley-Interscience, pp. 179–189, 2001.
- [25] "Continuous-time VCO," *Mathworks Documentation Center* [Online]. Available: <http://www.mathworks.com/help/comm/ref/continuoustimevco.html>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California